ENCICLOPEDIA PRACTICA DE LA

INFORMATICA APLICADA

16

El ordenador como instrumento musical y de composición

AIA

SINCLAIR AMSTRAD COMMODORE IBM



EDICIONES SIGLO CULTURAL

ENCICLOPEDIA PRACTICA DE LA

INFORMATICA APLICADA

16

El ordenador como instrumento musical y de composición

EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S.A.

Director-editor:
RICARDO ESPAÑOL CRESPO.

Gerente:
ANTONIO G. CUERPO.

Directora de producción:
MARIA LUISA SUAREZ PEREZ.

Directores de la colección:
MANUEL ALFONSECA, Doctor Ingeniero de Telecomunicación
y Licenciado en Informática
JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño y maquetación:
BRAVO-LOFISH.

Dibujos:
JOSE OCHOA Y ANTONIO PERERA.

Tomo XVI. **El ordenador como instrumento musical y de composición.**
AULA DE INFORMATICA APLICADA
ESTHER MALDONADO. Diplomada en solfeo y diplomada en arquitectura.

Ediciones Siglo Cultural, S.A.

Dirección, redacción y administración:
Sor Angela de la Cruz, 24-7.º G. Teléf. 279 40 36. 28020 Madrid.

Publicidad:
Gofar Publicidad, S.A. Benito de Castro, 12 bis. 28080 Madrid.

Distribución en España:
COEDIS, S.A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.
Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:
CADE, S.R.L. Pasaje Sud América. 1532. Teléf.: 21 24 64.
Buenos Aires - 1.290. Argentina.

ISBN del tomo: 84-7688-059-6
ISBN de la obra: 84-7688-018-9.

Fotocomposición:
ARTECOMP, S.A. Albarracín, 50. 28037 Madrid.

Imprime:
MATEU CROMO. Pinto (Madrid).

Depósito legal: M- 2.172-1987
Printed in Spain - Impreso en España.

Suscripciones y números atrasados:
Ediciones Siglo Cultural, S.A.
Sor Angela de la Cruz, 24-7.° G. Teléf. 279 40 36. 28020 Madrid

Octubre, 1986.
P.V.P. Canarias: 365,-

INDICE

Los programas que aparecen en este libro funcionan en los ordenadores:

IBM–PC, XT, AT y compatibles.
AMSTRAD–464, 664, 6128, 1512.
SINCLAIR–SPECTRUM 48 K, 128 K, PLUS, PLUS 2.
MSX–Todos los modelos.
COMMODORE–CBM 64 y CBM 128.

INTRODUCCION A LA MUSICA 1



AMOS a comenzar este capítulo y el libro con una introducción a la terminología y notación musical, que nos servirá para entender los capítulos posteriores, en los que veremos cómo convertir una pieza musical en un programa de ordenador o cómo transformar el teclado en un divertido piano.

Ante todo, no hay que dejarse abrumar por la notación musical. Aunque al principio pueda parecer muy críptica veremos que no es tan difícil como parece. Evidentemente, este capítulo no es un curso completo de solfeo, sino que tiene como objetivo familiarizar un poco al lector con el lenguaje musical.



EL PENTAGRAMA Y LAS NOTAS

Vamos a comenzar por ver de qué medios se valen los compositores para poner su obra sobre el papel.

La música se escribe sobre lo que se denomina *pentagrama,* que es una agrupación de cinco rayas tal y como muestra la figura 1.1.

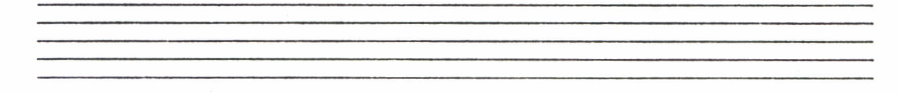


Fig. 1.1. Un pentagrama vacío.

Los diferentes tonos musicales se representan en el pentagrama mediante unos signos denominados *notas.* La altura de una nota en el penta-

grama indica su tono, de modo que cuanto más alta esté situada la nota, tendrá un tono más agudo, mientras que a medida que descienden, los tonos van siendo cada vez más graves.

Las notas se pueden escribir sobre las líneas del pentagrama o entre dos de ellas. La altura a la que se sitúa una nota en el pentagrama determina su nombre. En la figura 1.2, podemos ver un ejemplo. Hemos representado los nombres de las notas en español (DO, RE, MI, ...), así como en inglés (C, D, E, ...).

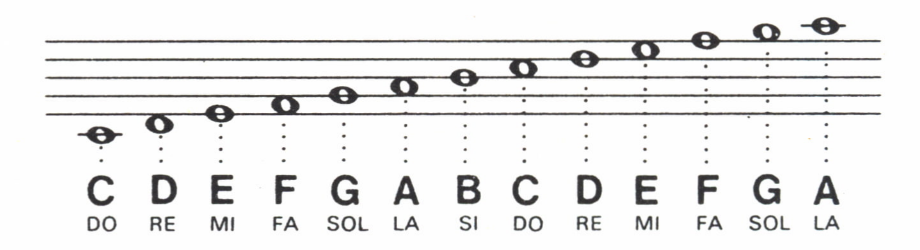


Fig. 1.2. Representación de las notas en el pentagrama.

Podemos observar que cada siete notas los nombres se repiten, es decir, después de DO, RE, MI, FA, SOL, LA, SI, la siguiente nota vuelve a ser DO. Sin embargo, este segundo DO está más alto en el pentagrama que el primer DO, por tanto el tono será más agudo, concretamente es una octava más alto (agudo) que el primer DO. Por el contrario el primer DO será una octava más bajo (grave) que el segundo DO. Si hay notas demasiado altas o demasiado bajas que no caben en el pentagrama, su altura se indica con unas pequeñas rayas adicionales que permiten ampliar el pentagrama. Sin embargo no conviene utilizar demasiadas rayas adicionales ya que esto dificultaría la lectura de la música.



LA DURACION

Ya sabemos representar el tono de una nota en el pentagrama, sin embargo, no todas las notas de una melodía duran lo mismo. Vamos a ver ahora cómo podemos representar su duración.

No debemos olvidar que las duraciones de las notas son siempre valores relativos ya que dependen del «tempo» o velocidad a la que se interpreta una composición, por lo tanto los valores del tiempo de las notas es-

tablecen su duración por comparación con otras. La figura 1.3 muestra los símbolos y duraciones de las notas, así como los nombres correspondientes a cada una.

Nota	*Nombre*	*Duración*
	Redonda	1
	Blanca	1/2
	Negra	1/4
	Corchea	1/8
	Semicorchea	1/16
	Fusa	1/32
	Semifusa	1/64

Fig. 1.3. Valores del tiempo de las notas.

La *redonda* es, normalmente, la nota más larga mientras que la *semifusa* es la más corta, teniendo una duración de 1/64 de la duración de la redonda, es decir, una redonda dura lo mismo que 64 semifusas. También podemos obtener duraciones intermedias entre los valores establecidos. Si escribimos un *puntillo* (.) a continuación de una nota, la duración aumentará en 1/2 con respecto a la duración que tenía. Por ejemplo, si una negra tiene una duración de 1/4 de redonda una negra con puntillo durará $1/4 + 1/2 \times 1/4 = 3/8$ con respecto a la duración de la redonda.

Por otra parte, en el pentagrama también podemos indicar períodos de silencio, durante los cuales no suena ninguna nota. Esto se consigue insertando pausas entre las notas. Estas pausas se representan con símbolos como los que se muestran en la figura 1.4. Las duraciones de estas pausas son las mismas que en las notas de su mismo nombre.



LAS CLAVES Y LAS ESCALAS

Antes dijimos que el pentagrama se podía ampliar con unas pequeñas líneas adicionales, sin embargo, para facilitar la lectura de la música se

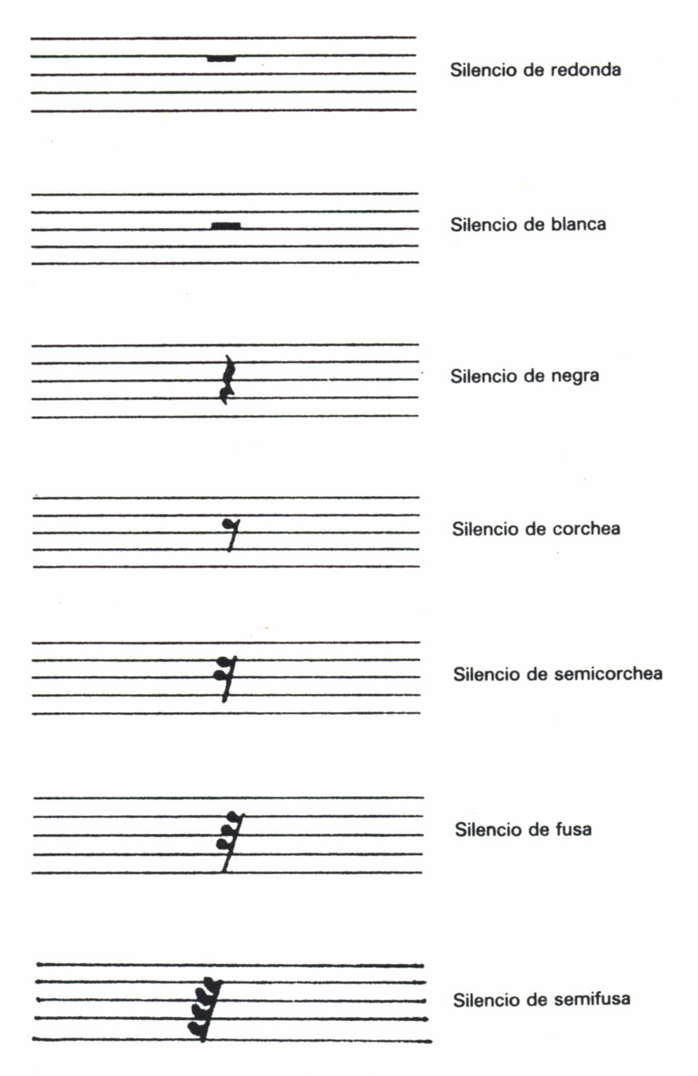


Fig. 1.4. Silencios.

pueden utilizar dos pentagramas paralelos. Para que esto resulte efectivo es necesario que el intervalo tonal de los dos pentagramas sea distinto. Para indicar el intervalo tonal de un pentagrama se utiliza un signo que se sitúa al principio del mismo y que se denomina *clave*. La clave fija el tono de una nota determinada. Las dos claves más utilizadas, sobre todo en composiciones para piano, son las de SOL y FA. La clave de SOL es la de tono más alto y fija el tono de la nota situada en la segunda línea del pentagrama, haciéndola corresponder con el SOL de la cuarta octava del piano. La

clave de FA, más grave, hace corresponder el tono de la nota situada en la cuarta línea del pentagrama con el FA de la tercera octava del piano.

Usando estas dos claves podemos abarcar fácilmente cuatro octavas tal y como muestra la figura 1.5.

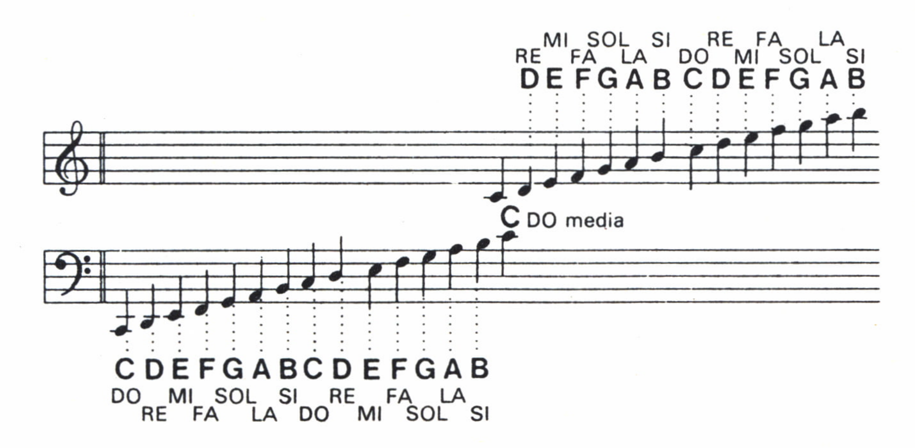


Fig. 1.5. Claves de SOL y FA.

Podemos observar que la posición de una nota determinada depende de la clave en la que esté. La nota DO media del piano queda entre los dos pentagramas.

Por otra parte, cada serie de ocho notas forma lo que se denomina una escala. La más sencilla es la de DO mayor que es la representada en la figura 1.6, así como su situación en el piano.

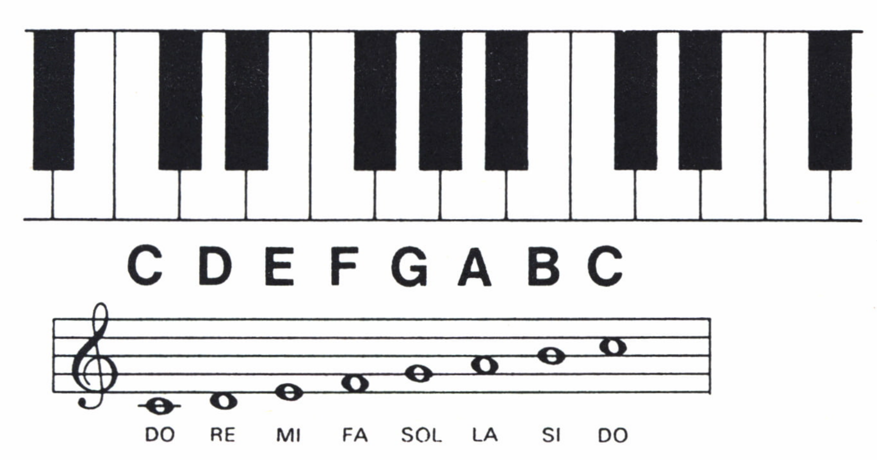


Fig. 1.6. Situación de las notas en las teclas del piano.

Podemos observar que la escala de DO mayor sólo utiliza las teclas blancas del piano. Sin embargo, entre las teclas blancas existen también teclas negras (excepto entre MI y FA y entre SI y DO). Si tocamos todas las teclas del piano (incluidas las negras) de izquierda a derecha, iremos avanzando de semitono en semitono. Después de tocar doce teclas habremos vuelto a la primera nota. Esta serie de doce semitonos se denomina escala cromática.

Todas las escalas mayores responden al mismo esquema tonal: tono-tono-semitono-tono-tono-tono-semitono, por tanto para ejecutar otras escalas mayores distintas de la de DO tendremos que utilizar las teclas negras del piano. A estas teclas se les designa por el nombre de cualquiera de las dos blancas adyacentes. Así por ejemplo, la tecla que está un semitono por encima de LA se llama LA *sostenido* (#), pero como también está un semitono por debajo de SI la podemos llamar SI *bemol* (b). Por tanto todas las notas con sostenido serán un semitono más altas de lo normal y todas las nota con bemoles serán un semitono más bajas. En la figura 1.7 podemos ver representados los nombres de todas las notas correspondientes a una escala cromática del piano.

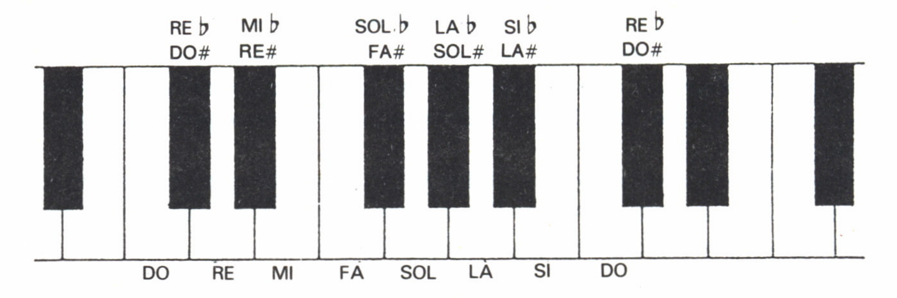


Fig. 1.7. Notas de la escala cromática en el piano.

Si ponemos sostenidos o bemoles al principio de una partitura, inmediatamente después de la clave, se les llama signatura de clave. Estos sostenidos y bemoles (conocidos en conjunto como alteraciones) se sitúan en las líneas o espacios del pentagrama correspondientes a la nota que se desea afectar durante toda la partitura. Así por ejemplo, un sostenido en la quinta línea del pentagrama en clave de SOL, o en la cuarta línea en clave de FA, indica que todas las notas FA que aparezcan en la partitura serán FA # y por tanto se elevarán siempre un semitono. En la figura 1.8 podemos observar algunos ejemplos de signaturas de clave.

El efecto de un sostenido o un bemol puede ser cancelado por el signo *becuadro* (♮) que devuelve la nota a su tono original.

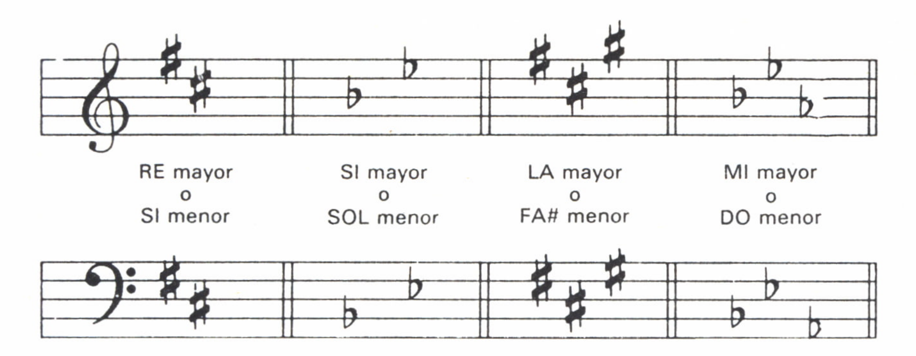


Fig. 1.8. Signaturas de clave.



LOS COMPASES Y EL RITMO

Los pentagramas se dividen en porciones, denominadas *compases,* que se indican con unas líneas verticales. Las líneas verticales dobles indican normalmente el final de un pasaje o de la pieza. Al principio de la partitura se indica la duración del compás (tiempo) mediante dos números, uno encima del otro. El número superior indica el número de movimientos o golpes de que consta el compás mientras que el inferior indica la duración de cada uno de ellos. En la figura 1.9 se muestra un ejemplo.

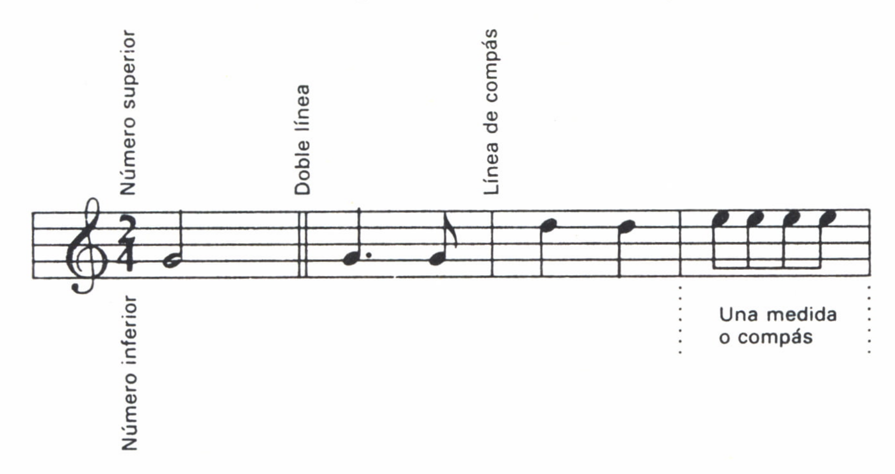


Fig. 1.9. Compases.

En este ejemplo el tiempo es 2/4, lo que significa que cada compás consta de dos golpes, cada uno de los cuales tiene una duración equivalen-

te a una negra representada por el 4 (recordemos que 4 negras equivalen a una redonda). Existen diferentes tiempos de compases lo que posibilita distintos *ritmos*.

Finalmente conviene señalar las diferentes formas en que se puede interpretar una nota. La nota en «*staccato*» se representa con un punto encima o debajo y se ejecuta recortando un poco su duración, dejando una breve pausa entre ella y la siguiente. El «*ligado*» es el efecto contrario. Se representa con un arco uniendo dos notas e indica que ambas se deben enlazar de forma suave y continua. Si el arco une dos notas del mismo tono, estará formando una «*ligadura*», en cuyo caso sólo se ejecutará una nota con una duración equivalente a la suma de las dos notas ligadas.

En la figura 1.10 podemos ver ejemplos de estas tres posibilidades.
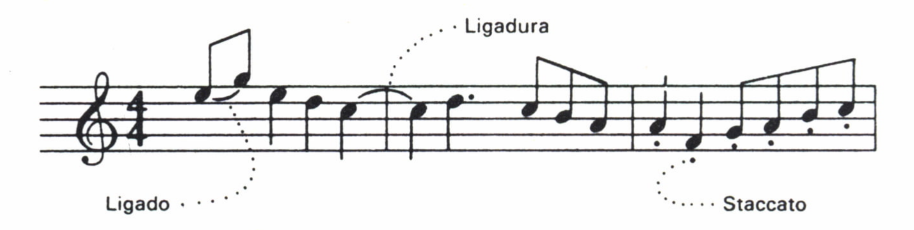


Fig. 1.10. Ejemplos de ligado, ligadura y staccato.



EL AIRE

El *aire* o *tempo* es el grado de lentitud o rapidez con que se debe interpretar una obra musical.

El aire se expresa mediante términos, generalmente italianos, que se colocan al principio de la partitura en la parte superior del pentagrama.

Los términos usados principalmente para expresar el aire son los siguientes:

Término	**Significado**	
Largo	Muy despacio	
Lento	Muy despacio	
Adagio	Despacio	↓ Velocidad creciente
Andante	Tranquilo	
Allegro	Aprisa	
Presto	Muy aprisa	
Vivace	Muy aprisa	
Vivo	Muy aprisa	

Por tanto el aire indica la velocidad de ejecución de una pieza musical.

EL SONIDO EN EL ORDENADOR 2



A lo largo del presente capítulo vamos a estudiar cómo es posible la generación de sonidos en distintos ordenadores. La mayoría de los ordenadores disponen de comandos BASIC específicos para la generación de sonidos como pueden ser **BEEP, SOUND** o **PLAY**. Sin embargo, la estructura de estos comandos suele variar bastante de un fabricante a otro, pudiendo producir desde sonidos sencillos hasta espectaculares efectos sonoros, dependiendo de la complejidad del chip de sonido que lleve incorporado cada ordenador. Por tanto vamos a dividir el capítulo en el estudio separado de los comandos de sonido en los principales ordenadores del mercado.



EL SONIDO EN EL SPECTRUM

En el **SPECTRUM**, la generación de sonidos se produce mediante la instrucción **BEEP** del **BASIC**, que tiene el siguiente formato:

BEEP D,F

donde el parámetro *D* es la duración del tono que va a sonar y se mide en segundos, y el parámetro *F* representa la frecuencia del sonido, expresada por un número entero comprendido en la gama que va desde –59 a +69. Cuanto menor sea el número de la frecuencia más grave será el tono del sonido y, a medida que aumentemos el valor de la frecuencia, aumentará el tono por toda la gama audible, pudiendo llegar a ser demasiado agudo para que lo capte el oído humano.

Por otra parte, y de cara a nuestras próximas experiencias musicales, conviene indicar que la nota **DO** media en las escalas del piano, está representada en el **SPECTRUM** por una frecuencia igual a 0. Si aumentamos

los valores de las frecuencias de uno en uno el tono del sonido producido aumentará de semitono en semitono. Lógicamente si disminuimos los valores de las frecuencias sucederá lo contrario.

Después de esta breve explicación pasemos ya a realizar nuestras primeras experiencias sonoras. Comencemos por ver cómo varía la duración de una nota con el parámetro D. El programa 2.1 hace sonar siempre la misma nota, el DO medio (frecuencia 0), pero va aumentando su duración cada vez que la ejecuta.

```
10 REM ***********************
20 REM * DURACION DEL SONIDO *
30 REM *      SPECTRUM       *
40 REM ***********************
50 LET D=0.01
60 FOR I=1 TO 10
70 PRINT AT 11,4;"DURACION: ";D;" SE
   GUNDOS"
80 BEEP D,0
90 PAUSE 25
100 LET D=D*2
110 NEXT I
```

Programa 2.1.

Conviene hacer notar que la duración D de la nota va expresada en segundos por tanto la primera nota que suena dura 0.01 segundos (una centésima de segundo) y la última 5.12 segundos. Sin embargo, las pausas producidas por la instrucción PAUSE no van expresadas en segundos sino según la equivalencia 50=1 segundo, lo que significa que la instrucción PAUSE de la línea 90 del programa produce una pausa de medio segundo entre sonido y sonido.

El programa 2.2 es una demostración de la gama completa de tonos disponible en el SPECTRUM.

```
10 REM *****************
20 REM * GAMA DE TONOS *
30 REM *   ESPECTRUM   *
40 REM *****************
50 FOR F=-59 TO 69
60 PRINT AT 11,11;"TONO: ";F;" "
70 BEEP 0.5,F
80 PAUSE 12.5
90 NEXT F
```

Programa 2.2.

Mediante un bucle **FOR-NEXT** hacemos variar el valor de la frecuencia desde –59, el tono más grave, a 69, el más agudo. Cada tono tiene una duración de medio segundo.

Sin embargo, si queremos utilizar el **SPECTRUM** para aplicaciones musicales muchos de estos tonos resultan demasiado graves o agudos. Un piano abarca una gama de tonos de siete octavas y cuarto (88 tonos) que corresponde a la gama de frecuencias que va desde –39 (LA) hasta 48 (DO). El programa 2.3 reproduce la escala cromática de la figura 2.1, a partir del DO medio.
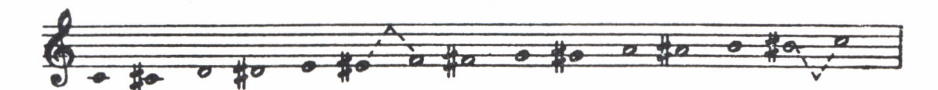


Fig. 2.1. Escala cromática.

```
10 REM ********************
20 REM * ESCALA CROMATICA *
30 REM *     SPECTRUM     *
40 REM ********************
50 FOR F=0 TO 11
60 READ N$
70 PRINT AT 11,15;N$;"   "
80 BEEP 0.5,F
90 PAUSE 5
100 NEXT F
110 DATA "DO","DO#","RE","RE#"
120 DATA "MI","FA","FA#","SOL"
130 DATA "SOL#","LA","LA#","SI"
```

Programa 2.3.

Los nombres de las notas que componen la escala cromática se almacenan en líneas **DATA** y se van imprimiendo en pantalla a medida que van sonando.

Sin embargo las escalas cromáticas suenan siempre un poco «raras». Por último el programa 2.4 hace que el ordenador interprete una escala en DO mayor.

```
10 REM **********************
20 REM * ESCALA EN DO MAYOR *
30 REM *      SPECTRUM      *
40 REM **********************
```

```
50 FOR I=1 TO 8
60 READ F,N$
70 PRINT AT 11,15;N$;" "
80 BEEP 0.5,F
90 PAUSE 5
100 NEXT I
110 DATA 0,"DO",2,"RE",4,"MI",5,"FA"
120 DATA 7,"SOL",9,"LA",11,"SI",12,"DO"
```

Programa 2.4.

En este caso es necesario almacenar en líneas DATA la frecuencia de cada tono (también almacenamos el nombre de cada nota), ya que al no ser el incremento del tono constante a lo largo de la escala, no se puede representar con el índice del bucle FOR-NEXT. Podemos observar que de MI (frecuencia 4) a FA (frecuencia 5) el incremento es de un semitono, al igual que de SI a DO, mientras que en el resto de las notas el incremento es de un tono.



EL SONIDO EN EL AMSTRAD

El AMSTRAD utiliza la instrucción SOUND del BASIC para la generación de sonidos. SOUND admite hasta siete parámetros de los cuales sólo los dos primeros son imprescindibles, mientras que los cinco restantes son opcionales. El formato completo de SOUND es el siguiente:

SOUND C, T, D, V, EV, ET, R

donde *C* indica la situación de los canales (el AMSTRAD dispone de tres canales de sonido); *T* es el período de tono, es decir, la forma técnica de especificar el tono del sonido, o lo que es lo mismo, la nota; *D* especifica la duración del sonido en centésimas de segundo (1=0.01 segundos); *V* expresa el volumen del sonido que se va a generar; *EV* es la envolvente de volumen que permite variar el volumen de una nota mientras está sonando; *ET* es la envolvente de tono que, análogamente a la envolvente de volumen, permite variar el tono de una nota mientras está sonando; por último *R* es el período de ruido que permite la generación de ruidos, es decir, sonidos poco musicales.

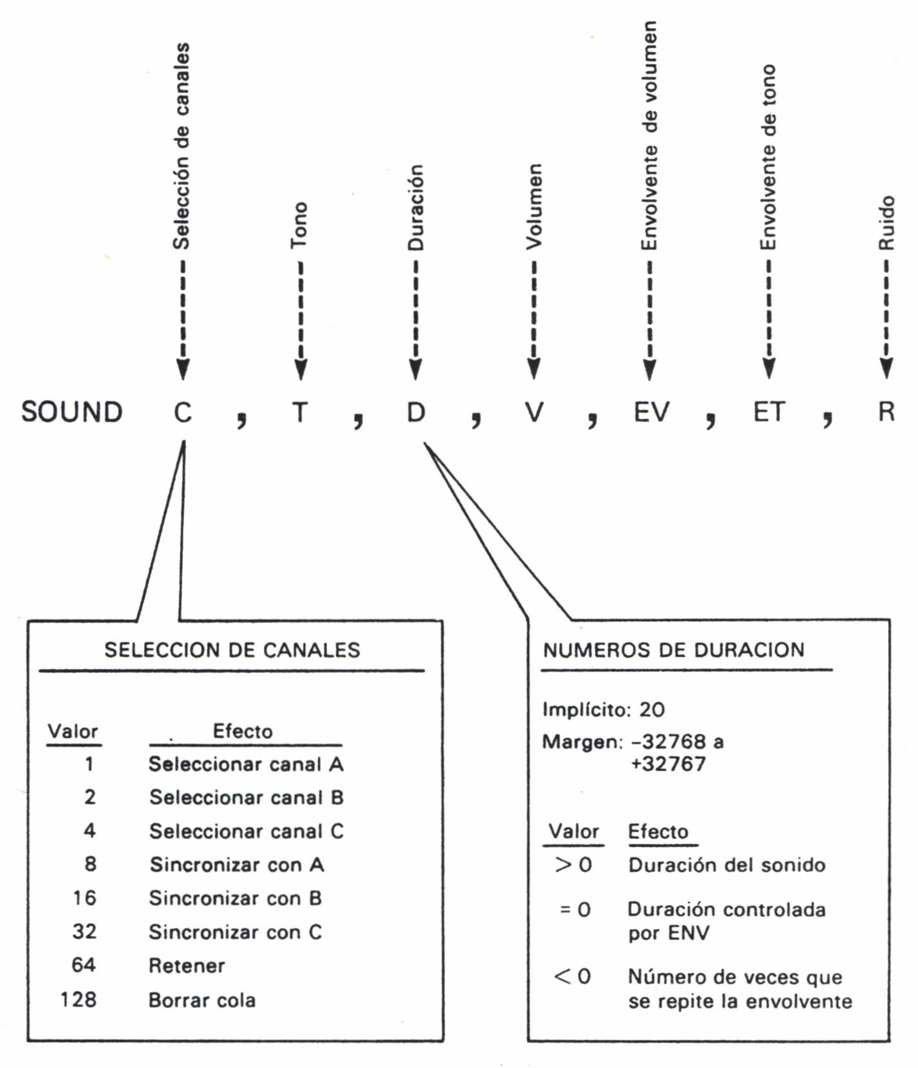


Fig. 2.2. Los parámetros de SOUND en AMSTRAD.

En este capítulo nos vamos a centrar en los cuatro primeros parámetros, dejando los restantes, de mayor complejidad, para capítulos posteriores.

El AMSTRAD dispone de tres canales de sonido: A, B y C que pueden actuar independientemente o simultáneamente produciendo acordes y otros efectos interesantes. Para ello el valor que se le puede asignar al parámetro C es un número entero comprendido entre 1 y 255, lo que nos da una idea de todas las situaciones de canales diferentes que podemos obte-

ner. Sin embargo, de momento sólo nos vamos a centrar en los primeros valores con los que se seleccionan los tres canales individualmente, tal y como se muestra en la figura 2.3.

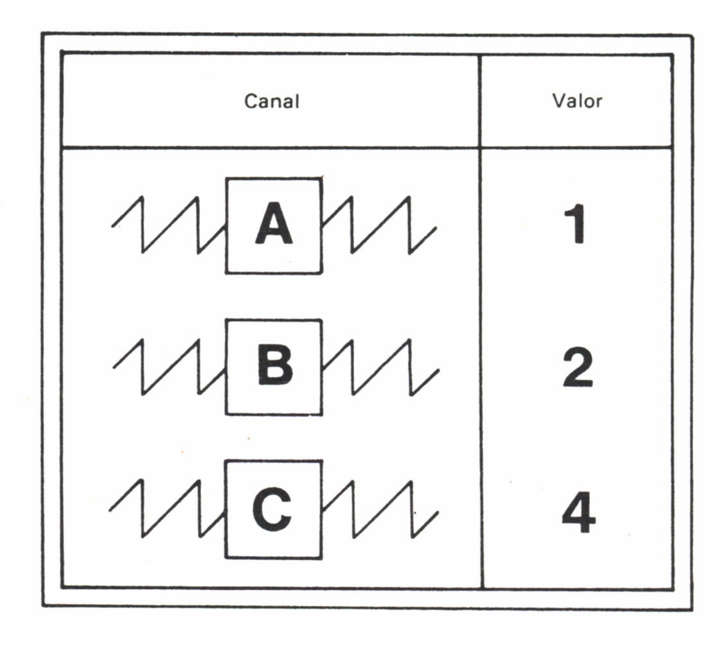


Fig. 2.3. Números de selección de canales en AMSTRAD.

De modo que, por ejemplo, la instrucción SOUND2 seleccionará el canal B para producir sonido, aunque aún falta, por lo menos, otro parámetro.

El segundo parámetro, T, define el tono del sonido en función de su período. A cada tono le corresponde una frecuencia y a cada frecuencia un período. El parámetro T puede tomar cualquier valor entero comprendido entre 0 (agudos) y 4095 (graves). El valor que utilicemos en cada caso puede corresponder a una nota situada dentro de las ocho octavas que puede abarcar el AMSTRAD. En la figura 2.4 podemos ver una tabla con los valores de los períodos de tono para cada una de las 96 notas que abarca el AMSTRAD, así como la frecuencia del tono y el error relativo producido al redondear el período a un valor entero (error sólo apreciable por los oídos más expertos).

NOTA		FRECUENCIA	PERIODO	ERROR RELATIVO	
C	DO	32.703	3822	-0.007%	
C#	DO#	34.648	3608	+0.007%	
D	RE	36.708	3405	-0.007%	
D#	RE#	38.891	3214	-0.004%	
E	MI	41.203	3034	+0.009%	
F	FA	43.654	2863	-0.016%	
F#	FA#	46.249	2703	+0.009%	Octava -3
G	SOL	48.999	2551	-0.002%	
G#	SOL#	51.913	2408	+0.005%	
A	LA	55.000	2273	+0.012%	
A#	LA#	58.270	2145	-0.008%	
B	SI	61.735	2025	+0.011%	

NOTA		FRECUENCIA	PERIODO	ERROR RELATIVO	
C	DO	65.406	1911	-0.007%	
C#	DO#	69.296	1804	+0.007%	
D	RE	73.416	1703	+0.022%	
D#	RE#	77.782	1607	-0.004%	
E	MI	82.407	1517	+0.009%	
F	FA	87.307	1432	+0.019%	
F#	FA#	92.499	1351	-0.028%	Octava -2
G	SOL	97.999	1276	+0.037%	
G#	SOL#	103.826	1204	+0.005%	
A	LA	110.000	1136	-0.032%	
A#	LA#	116.541	1073	+0.039%	
B	SI	123.471	1012	-0.038%	

NOTA		FRECUENCIA	PERIODO	ERROR RELATIVO	
C	DO	130.813	956	+0.046%	
C#	DO#	138.591	902	+0.007%	
D	RE	146.832	851	-0.037%	
D#	RE#	155.564	804	+0.058%	
E	MI	164.814	758	-0.057%	
F	FA	174.614	716	+0.019%	
F#	FA#	184.997	676	+0.046%	Octava -1
G	SOL	195.998	638	+0.037%	
G#	SOL#	207.652	602	+0.005%	
A	LA	220.000	568	-0.032%	
A#	LA#	233.082	536	-0.055%	
B	SI	246.942	506	-0.038%	

NOTA		FRECUENCIA	PERIODO	ERROR RELATIVO	
C	DO	261.626	478	+Ø.Ø46%	DO media
C#	DO#	277.183	451	+Ø.ØØ7%	
D	RE	293.665	426	+Ø.Ø81%	
D#	RE#	311.127	4Ø2	+Ø.Ø58%	
E	MI	329.628	379	-Ø.Ø57%	
F	FA	349.228	358	+Ø.Ø19%	
F#	FA#	369.994	338	+Ø.Ø46%	Octava Ø
G	SOL	391.995	319	+Ø.Ø37%	
G#	SOL#	415.3Ø5	3Ø1	+Ø.ØØ5%	
A	LA	44Ø.ØØØ	284	-Ø.Ø32%	LA internacional
A#	LA#	466.164	268	-Ø.Ø55%	
B	SI	493.883	253	-Ø.Ø38%	

NOTA		FRECUENCIA	PERIODO	ERROR RELATIVO	
C	DO	523.251	239	+Ø.Ø46%	
C#	DO#	554.365	225	-Ø.215%	
D	RE	587.33Ø	213	+Ø.Ø81%	
D#	RE#	622.254	2Ø1	+Ø.Ø58%	
E	MI	659.255	19Ø	+Ø.2Ø6%	
F	FA	698.457	179	+Ø.Ø19%	
F#	FA#	739.989	169	+Ø.Ø46%	Octava 1
G	SOL	783.991	159	-Ø.277%	
G#	SOL#	83Ø.6Ø9	15Ø	-Ø.328%	
A	LA	88Ø.ØØØ	142	-Ø.Ø32%	
A#	LA#	932.328	134	-Ø.Ø55%	
B	SI	987.767	127	+Ø.356%	

NOTA		FRECUENCIA	PERIODO	ERROR RELATIVO	
C	DO	1Ø46.5Ø2	119	-Ø.374%	
C#	DO#	11Ø8.731	113	+Ø.229%	
D	RE	1174.659	1Ø6	-Ø.39Ø%	
D#	RE#	1244.5Ø8	1ØØ	-Ø.441%	
E	MI	1318.51Ø	95	+Ø.2Ø6%	
F	FA	1396.913	89	-Ø.543%	
F#	FA#	1479.978	84	-Ø.548%	Octava 2
G	SOL	1567.982	8Ø	+Ø.35Ø%	
G#	SOL#	1661.219	75	-Ø.328%	
A	LA	176Ø.ØØØ	71	-Ø.Ø32%	
A#	LA#	1864.655	67	-Ø.Ø55%	
B	SI	1975.533	63	-Ø.435%	

NOTA		FRECUENCIA	PERIODO	ERROR RELATIVO	
C	DO	2Ø93.ØØ4	6Ø	+Ø.462%	
C#	DO#	2217.461	56	-Ø.662%	
D	RE	2349.318	53	-Ø.39Ø%	
D#	RE#	2489.Ø16	5Ø	-Ø.441%	
E	MI	2637.Ø21	47	-Ø.855%	
F	FA	2793.826	45	+Ø.574%	
F#	FA#	2959.955	42	-Ø.548%	Octava 3
G	SOL	3135.963	4Ø	+Ø.45Ø%	
G#	SOL#	3322.438	38	+Ø.992%	
A	LA	352Ø.ØØØ	36	+1.357%	
A#	LA#	3729.31Ø	34	+1.417%	
B	SI	3951.Ø66	32	+1.134%	

NOTA		FRECUENCIA	PERIODO	ERROR RELATIVO	
C	DO	4186.ØØ9	3Ø	+Ø.462%	
C#	DO#	4434.922	28	-Ø.662%	
D	RE	4698.636	27	+1.469%	
D#	RE#	4978.Ø32	25	-Ø.441%	
E	MI	5274.Ø41	24	+1.246%	
F	FA	5587.652	22	-1.685%	
F#	FA#	5919.911	21	-Ø.548%	Octava 4
G	SOL	6271.927	2Ø	+Ø.35Ø%	
G#	SOL#	6644.875	19	+Ø.992%	
A	LA	7Ø4Ø.ØØØ	18	+1.357%	
A#	LA#	7458.621	17	+1.417%	
B	SI	79Ø2.133	16	+1.134%	

Fig. 2.4. Tabla de notas, frecuencias y períodos de tono correspondientes.

Las frecuencias se calculan a partir de la correspondiente al LA internacional (frecuencia 440), mediante la fórmula siguiente:

FRECUENCIA = 440 * (2^ (OCTAVA + (N-10)/12))

donde N vale 1 para DO, 2 para DO1,... y 12 para SI.

A partir de la frecuencia se calcula el período mediante la fórmula:

PERIODO = ROUND (125000/FRECUENCIA)

Bueno pues ya podemos intentar algunas experiencias. El programa 2.5 reproduce una escala cromática a partir del DO medio por el canal A.

```
10 REM *********************
20 REM * ESCALA CROMATICA *
30 REM *      AMSTRAD      *
40 REM *********************
50 CLS
60 FOR I=1 TO 12
70 READ T,N$
80 LOCATE 19,13:PRINT N$;"  "
90 SOUND 1,T
100 FOR P=1 TO 500:NEXT
110 NEXT
120 DATA 478,DO,451,DO#,426,RE
130 DATA 402,RE#,379,MI,358,FA
140 DATA 338,FA#,319,SOL,301,SOL#
150 DATA 284,LA,268,LA#,253,SI
```

Programa 2.5.

Tanto los períodos como los nombres de las notas están almacenados en líneas **DATA**. El bucle **FOR-NEXT** de la línea 100 produce una pausa entre nota y nota. En este programa hemos utilizado únicamente los dos parámetros obligatorios para SOUND.

El parámetro D nos permite especificar la duración del sonido. Si lo omitimos el ordenador toma por defecto 20 centésimas de segundo. Efectivamente esta ha sido la duración de cada nota en el programa anterior (2.5). El valor máximo que podemos dar a D es de 32767. Experimentemos ahora con el programa 2.6.

```
10 REM ************************
20 REM * ESCALA Y DURACIONES *
30 REM *       AMSTRAD        *
40 REM ************************
50 CLS
60 D=2
70 FOR I=1 TO 8
80 READ T,N$
90 LOCATE 16,10:PRINT "NOTA: ";N$;"  "
100 LOCATE 10,16:PRINT "DURACION: ";D/100;" SEGUNDOS  "
110 SOUND 1,T,D
120 D=D*2
```

```
130 FOR P=1 TO 500:NEXT
140 NEXT
150 DATA 478,DO,426,RE
160 DATA 379,MI,358,FA
170 DATA 319,SOL,284,LA
180 DATA 253,SI,239,DO
```

Programa 2.6.

El programa permite al **AMSTRAD** ejecutar una escala en DO mayor con la particularidad de que cada nota dura el doble de la anterior, ya que la duración se multiplica por dos en la línea 120, por tanto el primer DO dura 0,02 segundos (línea 60) mientras que el segundo DO (octava nota) dura 2,56 segundos. El bucle **FOR-NEXT** de la línea 130 produce una pequeña pausa entre nota y nota. Finalmente, los tonos de las notas, así como sus nombres correspondientes, están almacenados en las líneas **DATA** del final del programa y su lectura se realiza en la línea 80.

Por último, el parámetro V, nos permite asignar un volumen distinto a cada sonido. El valor que puede tomar V es un número entero comprendido entre 0 y 15, siendo 0 el mínimo y 15 el máximo. Cuando no se específica ningún valor para el volumen, el **AMSTRAD** toma por defecto el 12. Para observar las variaciones de volumen podemos probar a ejecutar el programa 2.7.

```
10 REM **********************
20 REM * ESCALAS Y VOLUMEN *
30 REM *      AMSTRAD       *
40 REM **********************
50 CLS
60 FOR V=1 TO 15 STEP 2
70 GOSUB 200
80 NEXT
90 FOR V=15 TO 1 STEP -2
100 GOSUB 200
110 NEXT
120 END
200 READ T,N$
210 LOCATE 16,10:PRINT "NOTA: ";N$;"  "
220 LOCATE 16,16:PRINT "VOLUMEN: ";V;"  "
230 SOUND 1,T,50,V
240 FOR P=1 TO 500:NEXT
250 RETURN
```

```
300 DATA 239,DO,213,RE
310 DATA 190,MI,179,FA
320 DATA 159,SOL,142,LA
330 DATA 127,SI,119,DO
340 DATA 119,DO,127,SI
350 DATA 142,LA,159,SOL
360 DATA 179,FA,190,MI
370 DATA 213,RE,239,DO
```

Programa 2.7.

Este programa ejecuta una escala en DO mayor ascendente en la cual va aumentando el volumen progresivamente gracias al bucle **FOR-NEXT** de las líneas 60-80. A continuación ejecuta la misma escala pero en sentido inverso (descendente) a la vez que va disminuyendo el volumen mediante el bucle decreciente de las líneas 90-110. Tanto los tonos como los nombres de cada nota se almacenan en líneas **DATA**. La subrutina de las líneas 200-250 se encarga de realizar la lectura de cada nota, así como de imprimirla en pantalla y generar su sonido.

Finalmente, el programa 2.8 nos permite probar distintos sonidos del **AMSTRAD** seleccionando el tono, la duración y el volumen.

```
10 REM ****************************
20 REM * EXPERIENCIAS CON SONIDOS *
30 REM *          AMSTRAD         *
40 REM ****************************
50 CLS
60 INPUT "TONO (0-4095) ";T
70 INPUT "DURACION (1-32767) ";D
80 INPUT "VOLUMEN (0-15) ";V
90 CLS
100 LOCATE 16,10:PRINT "TONO: ";T
110 LOCATE 16,12:PRINT "DURACION: ";D
120 LOCATE 16,14:PRINT "VOLUMEN: ";V
130 SOUND 1,T,D,V
140 LOCATE 1,22:PRINT "PULSA UNA TECLA PARA HACER OTRA
PRUEBA
150 IF INKEY$="" THEN GOTO 150 ELSE GOTO 50
```

Programa 2.8.



EL SONIDO EN EL IBM

El IBM dispone únicamente de un canal de sonido, sin embargo tiene la posibilidad de generar sonidos con dos instrucciones distintas: SOUND y PLAY. Vamos a comenzar por la primera, que es más sencilla. La instrucción SOUND tiene el siguiente formato:

SOUND F,D

donde el parámetro *F* es la frecuencia del sonido expresada en Herzios. Debe ser una expresión numérica perteneciente al rango de 37 a 32767. Cuanto mayor sea el valor de la frecuencia más alto será el tono del sonido generado.

Para saber las frecuencias correspondientes a las ocho octavas utilizadas normalmente en música podemos consultar la tabla de la figura 2.4. Por otra parte, el parámetro D es la duración del sonido expresada en tic-tac de reloj, sabiendo que 1 segundo equivale a 18,2 tic-tac. La duración debe ser una expresión numérica comprendida entre 0 y 65535.

La sentencia SOUND, produce un sonido y continúa la ejecución del programa hasta que llega a una nueva sentencia SOUND. Si la duración de la nueva sentencia SOUND es 0 la sentencia actual SOUND, que se está ejecutando, se desconecta. En cualquier otro caso el programa espera a que finalice el primer sonido antes de ejecutar la nueva sentencia SOUND.

Para crear períodos de silencio podemos utilizar SOUND 32767,D, ya que la frecuencia 32767 es demasiado aguda para ser percibida por el oído humano.

En la figura 2.5 podemos ver una tabla que muestra los «tempos» musicales típicos en términos de tic-tac de reloj.

El programa 2.9 es un ejemplo del funcionamiento de SOUND, en el que varía simultáneamente el tono y la duración.

La línea 100 produce un período de silencio entre nota y nota. En líneas DATA están almacenados los nombres de las notas correspondientes a una escala en DO mayor, así como sus frecuencias asociadas.

Por otra parte, el IBM también puede generar sonidos mediante la sentencia PLAY. Esta instrucción está más asociada con la música que SOUND, ya que sólo permite ejecutar los tonos correspondientes a las siete octavas habituales. El formato de PLAY es el siguiente:

PLAY cadena

donde *cadena* es una expresión de caracteres que consiste en mandatos de música de un solo carácter. PLAY introduce un «lenguaje de definición de melodía» en una cadena de caracteres.

Tempo		Pulsa-ciones/ Minuto	Tic-tac/ Pulsación
muy lento	Muy largo		
	Largo	4∅-6∅	27.3-18.2
	Larguetto	6∅-66	18.2-16.55
	Grave		
	Lento		
	Adagio	66-76	16.55-14.37
lento	Adagietto		
	Andante	76-1∅8	14.37-1∅.11
moderado	Andantino		
	Moderato	1∅8-12∅	1∅.11-9.1
	Allegro	12∅-168	9.1-6.5
	Vivace		
	Veloce		
muy	Presto	168-208	6.5-5.25
rápido	Prestissimo		

Fig. 2.5. Tempos expresados en tic-tac de reloj.

```
10 REM **************************
20 REM * LOS SONIDOS CON SOUND *
30 REM *          IBM          *
40 REM **************************
50 KEY OFF:WIDTH 40:CLS
60 FOR D=5 TO 40 STEP 5
70 READ F,N$
80 LOCATE 12,19:PRINT N$;"   "
90 SOUND F,D
100 SOUND 32767,5
110 NEXT
120 DATA 1046.50,DO,987.77,SI
130 DATA 880.00,LA,783.99,SOL
140 DATA 698.46,FA,659.26,MI
150 DATA 587.33,RE,523.25,DO
```

Programa 2.9.

Los mandatos de música de un solo carácter que se pueden emplear en PLAY son los siguientes:

De A a G: Emite la nota indicada en la octava actual, teniendo en cuenta la correspondencia entre notación inglesa y española

(DO=C, RE=D, ... LA=A, SI=B). Después de la nota se puede añadir un signo # o + que indican un sostenido, o un signo – que indica un bemol. Estos signos no se pueden emplear si la nota no corresponde a una tecla negra del piano, por ejemplo E# no sería válido (MI# es lo mismo que FA). Ejemplo: PLAY «CDEFGAB»

On: Carga la octava deseada. Hay 7 octavas numeradas de 0 a 6, que son los valores que puede tomar n. La octava 3 es la que comienza con el DO central (C). La octava 4 se asume por defecto. Ejemplo: PLAY «03 CDE 04 CDE 05 CDE».

>n: Eleva una octava y toca la nota especificada en n.

<n: Desciende una octava y toca la nota especificada en n.

Nn Toca la nota n. El parámetro n puede tomar cualquier valor entero comprendido entre 0 y 84. El valor 0 indica silencio, mientras que el resto de los valores corresponden a las 84 notas que abarcan las siete octavas. Es una forma alternativa de selección de notas, además de la ya explicada de selección de octava (On) y nota (A-G). Ejemplo: PLAY «N49 N51 N53».

Ln: Define la duración de las notas siguientes. La duración real (relativa) es 1/n. El parámetro n puede estar en el rango de 1 a 64, con la equivalencia siguiente:

L1 .... redonda
L2 .... blanca
L4 .... negra
L8 .... corchea
L16 .... semicorchea
L32 .... fusa
L64 .... semifusa

aunque también se pueden asignar valores intermedios. Si sólo se quiere variar la duración de una nota, se puede especificar su valor a continuación de dicha nota. Por ejemplo A8 es lo mismo que L8A. Ejemplo: PLAY "«C1 D2 E4 F8 G16 A32 B64».

Pn: Representa la duración de un silencio del mismo modo que Ln.

.: Puntillo. Se puede colocar después de una nota o de un silencio, produciendo los efectos explicados en el capítulo 1. Se puede poner más de un puntillo, por ejemplo, una blanca con dos puntillos tendrá una duración equivalente a una blanca más una negra más una corchea.

Tn: Define el «tempo» (aire). El valor de n debe estar comprendido entre 32 y 255.

MF: Música de primer plano. Cada nota no empezará hasta que

no haya concluido la anterior. MF es el estado asumido por el ordenador por omisión.

MB: Música de fondo. Cada nota se coloca en una memoria intermedia que permite que continúe la ejecución del programa en BASIC mientras suena la música.

MN: Música normal. Cada nota suena 7/8 de la duración especificada con Ln. Este es el estado que el ordenador toma por defecto para MN, ML y MS.

ML: Ligado. Cada nota suena el total de su duración.

MS: Staccato. Cada nota suena 3/4 de la duración especificada con Ln.

X variable: Ejecuta la cadena de caracteres especificada en la variable. En este caso es obligatorio poner un punto y coma tras la variable.

En todos estos mandatos el parámetro *n* puede ser una constante o, con el signo =, una variable.

El programa 2.10 ejecuta la escala en DO mayor descendente desde la octava 6 hasta la 0.

```
10 REM *************************
20 REM * ESCALAS DESCENDENTES *
30 REM *          IBM          *
40 REM *************************
50 KEY OFF:WIDTH 40:CLS
60 LOCATE 13,10:PRINT "ESCALAS DESCENDENTES"
70 ESCALA$="BAGFEDC"
80 PLAY "O6 XESCALA$;"
90 FOR I=1 TO 6
100 PLAY "<XESCALA$;"
110 NEXT
```

Programa 2.10.

En la línea 70 asignamos a la variable ESCALA$ una cadena musical compuesta por las 7 notas que componen una escala descendente (de SI a DO).

La línea 80 ejecuta dicha escala en la octava 6. A continuación, mediante un bucle FOR-NEXT que se repite 6 veces se va repitiendo la escala pero cada vez una octava más baja, hasta alcanzar la octava 0.

En capítulos posteriores veremos más en profundidad todos estos mandatos de modo que podamos sacarles el máximo rendimiento.



EL SONIDO EN EL COMMODORE

La generación de sonidos en el **COMMODORE** puede parecer, en un principio, más compleja que en los demás ordenadores ya que no dispone de ninguna instrucción **BASIC** específica para el sonido. Sin embargo, el **COMMODORE** cuenta con un chip de sonido, el **SID**, con características similares al del **AMSTRAD**, es decir, puede disponer de hasta tres canales de sonido, a la vez que libera al ordenador para que realice otras tareas simultáneamente.

Otra peculiaridad del **COMMODORE** es que, a diferencia de los demás ordenadores, no cuenta con un altavoz incorporado por lo que los sonidos los emite a través del altavoz del televisor.

El SID cuenta con 29 registros que van desde la posición de memoria 54272 hasta la 54300. Cada uno de estos registros tiene una función específica que iremos viendo poco a poco. A grandes rasgos podemos decir que cada uno de los tres canales de sonido se controla mediante siete registros. De los ocho registros restantes, cuatro de ellos tienen funciones que afectan a la vez a las tres voces y los otros cuatro funciones especiales. Cada vez que queramos producir un sonido determinado deberemos actuar sobre algunas de estas direcciones de memoria de la forma siguiente:

POKE DM, V

donde *DM* es la dirección de memoria sobre la que se actúa, y el parámetro *V* es el valor numérico que se asigna a dicha posición de memoria.

Pasemos a estudiar ahora la función de cada una de estas posiciones de memoria. En primer lugar, antes de realizar ninguna operación, es conveniente «limpiar» todos los registros de sonido, es decir, ponerlos a cero, de lo contrario podrían aparecer sonidos inesperados. Esto se consigue

```
10 REM ****************************
20 REM * LIMPIEZA DE REGISTROS *
30 REM *        COMMODORE          *
40 REM ****************************
50 PRINT CHR$(147)
60 FOR I=54272 TO 54296
70 POKE I,0
80 NEXT I
```

Programa 2.11.

con un sencillo bucle como el que muestra el programa 2.11, que recorre cada una de las posiciones de memoria.

Este bucle lo pondremos siempre al principio de nuestros programas musicales. Después de esto hay que definir el nivel de volumen deseado. La dirección correspondiente al volumen es la 54296 y el nivel de volúmen que podemos asignar es un valor entero comprendido entre 0 (mínimo) y 15 (máximo). La siguiente instrucción ajustaría el volumen al máximo.

```
POKE 54296,15
```

Tras el volumen deben establecerse los parámetros del modulador de amplitud para cada una de las voces que se van a usar. Estos parámetros definen la denominada *envolvente de volumen* de un sonido y se puede representar con una curva como la de la figura 2.6.

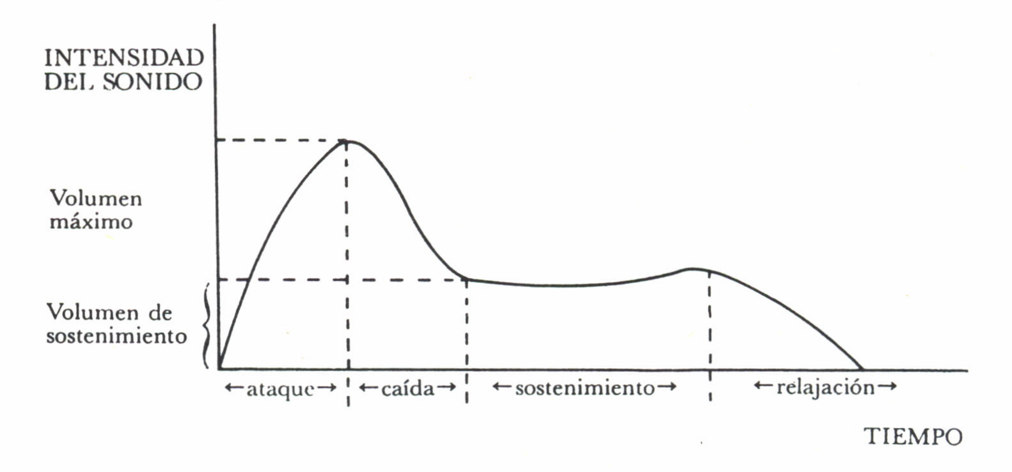


Fig. 2.6. Envolvente de volumen de un sonido.

Podemos observar que la intensidad de una nota va cambiando mientras suena. Durante el primer período, denominado de *ataque,* el volumen del sonido crece hasta el nivel máximo. A continuación empieza a descender, en el período denominado *caída,* hasta un cierto nivel de mantenimiento. El tercer período, denominado de *sostenimiento,* el volumen se mantiene hasta que alcanza el período de *relajación* durante el cual el sonido se extingue. Para introducir estos datos tenemos que seleccionar el nivel de sostenimiento a utilizar que debe estar comprendido entre 0 y 15 y los valores de ataque, caída y relajación correspondientes a los tiempos que se requieran según la tabla de la figura 2.7.

Valor	Tiempo de ataque	Tiempo de caída y relajación
0	2 ms	6 ms
1	8 ms	24 ms
2	16 ms	48 ms
3	24 ms	72 ms
4	38 ms	114 ms
5	56 ms	168 ms
6	68 ms	204 ms
7	80 ms	240 ms
8	100 ms	300 ms
9	250 ms	750 ms
10	500 ms	1.5 s
11	800 ms	2.4 s
12	1 s	3 s
13	3 s	9 s
14	5 s	15 s
15	8 s	24 s

Fig. 2.7. Tabla de valores de ataque, caída y relajación.

Los valores seleccionados se introducen en dos registros, uno de ataque-caída y otro de sostenimiento-relajación. Las direcciones respectivas de estos dos registros son 54277 y 54278 para el canal 1, 54284 y 54285 para el canal 2 y 54291 y 54292 para el canal 3. Para introducir los valores, por ejemplo, en el canal 1, podemos utilizar las siguientes sentencias:

```
POKE 54277, A*16+C
POKE 54278, S*16+R
```

siendo *A* el valor seleccionado para el ataque, *C* el de caída, *S* el de sostenimiento y *R* el de relajación.

A continuación, hay que establecer la frecuencia del sonido que se desea producir. Para ello hay que fijar dos valores denominados parte baja de la frecuencia y parte alta. Estos dos valores se introducen en sendos registros que son respectivamente los de direcciones 54272 y 54273 para la voz 1, 54279 y 54280 para la voz 2, y 54286 y 54287 para la voz 3. Por ejemplo las instrucciones:

```
POKE 54272, 37
POKE 54273, 17
```

introducen los valores de frecuencia correspondientes el DO medio en el canal 1.

El COMMODORE puede abarcar ocho octavas equivalentes a las utilizadas habitualmente en música. Los valores de altas y bajas frecuencias correspondientes a las 96 notas están representados en la tabla de la figura 2.8.

Nota	*Octava*	*Alta frecuencia*	*Baja frecuencia*
DO	0	1	18
DO sostenido	0	1	35
RE	0	1	52
RE sostenido	0	1	70
MI sostenido	0	1	90
FA sostenido	0	1	110
FA sostenido	0	1	132
SOL	0	1	155
SOL sostenido	0	1	179
LA	0	1	205
LA sostenido	0	1	133
SI	0	2	6
DO	1	2	37
DO sostenido	1	2	69
RE	1	2	104
RE sostenido	1	2	140
MI	1	2	179
FA sostenido	1	2	220
FA	1	2	8
SOL	1	3	54
SOL sostenido	1	3	103
LA	1	3	155
LA sostenido	1	2	210
SI	1	4	12
DO	2	4	73
DO sostenido	2	4	139
RE	2	4	208
RE sostenido	2	5	25
MI	2	5	103
FA	2	5	185
FA sostenido	2	6	16
SOL	2	6	108
SOL sostenido	2	6	206
LA	2	7	53
LA sostenido	2	7	163
SI	2	8	23
DO	3	8	147
DO sostenido	3	9	21
RE	3	9	159
RE sostenido	3	10	60

Nota	*Octava*	*Alta frecuencia*	*Baja frecuencia*
MI	3	10	205
FA	3	11	114
FA sostenido	3	12	32
SOL	3	12	216
SOL sostenido	3	13	156
LA	3	14	107
LA sostenido	3	15	70
SI	3	16	47
DO	4	17	37
DO sostenido	4	18	42
RE	4	19	63
RE sostenido	4	20	100
MI	4	21	154
FA	4	22	227
FA sostenido	4	24	63
SOL	4	25	177
SOL sostenido	4	27	56
LA	4	28	214
LA sostenido	4	30	141
SI	4	32	94
DO	5	34	75
DO sostenido	5	36	85
RE	5	38	126
RE sostenido	5	40	200
MI	5	43	52
FA	5	45	198
FA sostenido	5	48	127
SOL	5	51	97
SOL sostenido	5	54	111
LA	5	57	172
LA sostenido	5	61	126
SI	5	64	188
DO	6	68	149
DO sostenido	6	72	169
RE	6	76	252
RE sostenido	6	81	161
MI	6	86	105
FA	6	91	140
FA sostenido	6	96	254
SOL	6	102	194

Nota	*Octava*	*Alta frecuencia*	*Baja frecuencia*
SOL sostenido	6	108	223
LA	6	115	88
LA sostenido	6	122	52
SI	6	129	120
DO	7	137	43
DO sostenido	7	145	83
RE	7	153	247
RE sostenido	7	163	31
MI	7	172	210
FA	7	183	25
FA sostenido	7	193	252
SOL	7	205	133
SOL sostenido	7	217	189
LA	7	230	176
LA sostenido	7	244	103
SI	7	253	46

Fig. 2.8. Tabla de notas y sus frecuencias para COMMODORE.

Finalmente, es necesario definir la forma de la onda que se va a producir. El COMMODORE dispone de cuatro formas de onda posibles. Las direcciones de memoria correspondientes a la forma de onda son: 54276 para la primera voz, 54283 para la segunda y 54290 para la tercera. Los nombres y valores que se asignan a cada forma de onda están recogidos en la figura 2.9.

Nombre	*Comienzo de la nota*	*Fin de la nota*
Triangular	17	16
Diente de sierra	33	32
Cuadrada	65	64
Ruido	129	128

Fig. 2.9. Formas de onda para el COMMODORE.

Podemos observar que para cada forma de onda se recogen dos valores. La razón es que cuando se introduce el primer valor (comienzo de la nota) en el registro correspondiente se inicia el ciclo de la envolvente en el que el período de sostenimiento durará hasta que el registro tome el va-

lor de fin de la nota. En este momento comenzará el período de relajación según el valor elegido.

Bueno, con todo lo visto ya podemos empezar a experimentar con algún programa sonoro. Por ejemplo el programa 2.12 nos permite escuchar los distintos efectos del modulador de amplitud al variar los valores de ataque, caída, sostenimiento y relajación.

```
10 REM **************************
20 REM * MODULADOR DE AMPLITUD *
30 REM *        COMMODORE        *
40 REM **************************
50 FOR I=54272! TO 54296!
60 POKE I,0
70 NEXT I
80 POKE 54296!,15
90 A=6:D=6:S=6:R=6
100 PRINT CHR$(147)
110 PRINT "ATAQUE","CAIDA","SOSTEN","RELAX",
A,D,S,R
120 FOR I=1 TO 8
130 POKE 54277!,A*16+D
140 POKE 54278!,S*16+R
150 READ BF,AF
160 POKE 54272!,BF:POKE 54273!,AF
170 POKE 54276!,33
180 FOR T=1 TO 100:NEXT T
190 POKE 54276!,32
200 NEXT I
210 PRINT SPC(200)
220 PRINT "PULSA UNA TECLA DE FUNCION PARA
VARIAR LOS VALORES DE LA ENVOLVENTE"
230 GET A$:IF A$="" THEN GOTO 230
240 IF ASC(A$)<133 OR ASC(A$)>140 THEN
GOTO 230
250 ON ASC(A$)-132 GOSUB 300,320,340,360,
380,400,420,440
260 RESTORE
270 GOTO 100
280 DATA 75,35,126,38,52,43,198,45
290 DATA 97,51,172,57,188,64,149,68
300 A=A+1:IF A>15 THEN A=6
310 RETURN
320 D=D+1:IF D>15 THEN D=6
330 RETURN
340 S=S+1:IF S>15 THEN S=6
350 RETURN
360 R=R+1:IF R>15 THEN R=6
```

```
370 RETURN
380 A=A-1:IF A<O THEN A=6
390 RETURN
400 D=D-1:IF D<O THEN D=6
410 RETURN
420 S=S-1:IF S<O THEN S=6
430 RETURN
440 R=R-1:IF R<O THEN R=6
450 RETURN
```

Programa 2.12.

El bucle FOR-NEXT de las líneas 50-70 pone todos los registros de sonido a cero. A continuación, en la línea 80 ajustamos el volumen máximo. En la línea 90 establecemos los valores iniciales de ataque, caída, sostenimiento y relajación. El bucle FOR-NEXT de las líneas 120-200 interpreta una escala en DO mayor con los valores establecidos para la envolvente. En la línea 130 se introduce el valor de ataque-caída para el canal 1. Asimismo en la línea 140 se introduce el valor de sostenimiento-relajación, también para el canal 1.

La línea 160 introduce los valores de baja frecuencia y alta frecuencia correspondientes en los registros asignados al canal 1. En la línea 170 establecemos una forma de onda en diente de sierra. La nota sonará mientras se ejecuta el bucle FOR-NEXT de la línea 180. A continuación la línea 190 marca el final de la nota. Finalmente podemos variar los valores de la envolvente, aumentando o disminuyendo cada uno de ellos, según la tecla de función (F1 ... F8) que pulsemos. La escala musical volverá a sonar con la nueva envolvente definida. Por último, las líneas DATA almacenan los valores de bajas y altas frecuencias de las ocho notas correspondientes a la quinta octava.

El programa 2.13 nos permite apreciar los diferentes timbres que podemos obtener variando la forma de la onda.

```
10 REM *****************
20 REM * FORMA DE ONDA *
30 REM *   COMMODORE   *
40 REM *****************
50 FOR I=54272 TO 54296
60 POKE I,0
70 NEXT I
80 POKE 54296,15
```

```
90 POKE 54277,68:POKE 54278,132
100 FO=16
110 PRINT CHR$(147)
120 FOR I=1 TO 8
130 READ BF,AF
140 POKE 54272,BF:POKE 54273,AF
150 POKE 54276,FO+1
160 FOR T=1 TO 100:NEXT T
170 POKE 54276,FO
180 NEXT I
190 FO=FO*2
200 IF FO>128 THEN END
210 RESTORE
220 GOTO 120
230 DATA 149,68,188,64,172,57,97,51
240 DATA 198,45,52,43,126,38,75,34
```

Programa 2.13.

Al ejecutarlo podremos escuchar una escala en DO mayor descendente que se repetirá cuatro veces con distintas formas de onda.

Por último, en la figura 2.10 podemos ver una tabla-resumen de los registros del SID y sus funciones. La columna SID+ representa los valores que habría que sumar a la variable SID (suponiendo SID=54272) para obtener la dirección de memoria adecuada.



EL SONIDO EN MSX

El sistema MSX posee un chip de sonido, el PSG, que podemos controlar desde el BASIC mediante la instrucción SOUND. El PSG consta de 16 registros, cada uno de los cuales almacena la información de una característica del sonido. De estos 16 registros nosotros sólo podemos utilizar los 14 primeros numerados del 0 al 13. La instrucción SOUND nos permite alterar el valor de estos registros SOUND tiene el siguiente formato:

SOUND R,CS

siendo *R* el número de registro sobre el que vamos a actuar (0-13) y *CS* la característica sonora que aporta dicho registro, pudiendo ser cualquier valor comprendido entre 0 y 255.

Veamos ahora el significado de cada registro. El PSG puede producir

Dirección			SID+			Función	Valor en el registro
1	2	3	1	2	3		
54272	54279	54286	Ø	7	14	Parte baja de la frecuencia	Ø-255 (tabla)
54273	5428Ø	54287	1	8	15	Parte alta de la frecuencia	Ø-255 (tabla)
54274	54281	54288	2	9	16	Parte baja de ancho de pulso	Ø-255
54275	54282	54289	3	1Ø	17	Parte alta de ancho de pulso	Ø-15
54276	54283	5429Ø	4	11	18	Forma de onda	Forma + 1 si se pone / Ø si se quita
54277	54284	54291	5	12	19	Ataque-caída	16•Ataque+caída
54278	54285	54292	6	13	2Ø	Sostenimiento-relajación	16•sostenimiento+relajación
	54293			21		Parte baja de la frecuenc. de corte	Ø-7
	54294			22		Parte alta de la frecuenc. de corte	Ø-255
	54295			23		Resonancia y filtros	•1 si filtra voz 1 Reson. (Ø-15•16+ •2 si filtra voz 2 •4 si filtra voz 4
	54296			24		Tipo de filtro y volumen	64 Si HP 32 Si BP+VOL (Ø-15) 16 Si LP (Ø-15)
	54297			25		Potenciómetro 1	
	54298			26		Potenciómetro 2	Leídos
	54299			27		Oscilador 3	con
	543ØØ			28		Envolvente 3	PEEK

Fig. 2.10. Tabla resumen del SID del COMMODORE.

simultáneamente tres señales de tono y una de ruido las cuales se envían al altavoz a través de los canales A, B y C.

El tono de cada canal se determina cargando los valores de frecuencia deseados en los registros del 0 al 5 (0-1 canal A, 2-3 canal B y 4-5 canal C). El registro 6 se utiliza para cargar la frecuencia de ruido. El registro 7 sirve para seleccionar los canales que emiten tono y los que emiten ruido. Los 3 canales pueden emitir tono y ruido simultáneamente o por separado. El volumen de cada canal se ajusta en los registros 8, 9 y 10 (canales A, B y C respectivamente). Los registros 11 y 12 se utilizan para controlar el período de la envolvente de la onda sonora. Finalmente el registro 13 define el tipo de envolvente que vamos a utilizar. La figura 2.11 muestra una tabla esquemática de estos registros.

Hemos visto que cada canal utiliza dos registros para seleccionar el tono. Realmente de estos dos registros sólo se utilizan 12 bits, los 8 bits me-

N.º Registro	Función	b7	b6	b5	b4	b3	b2	b1	b0
0	Frecuencia canal A	FT (A)							
1						CT (A)			
2	Frecuencia canal B	FT (B)							
3						CT (B)			
4	Frecuencia canal C	FT (C)							
5						CT (C)			
6	Frecuencia ruido				NP				
7	Selección canal salida	1	0	Ruido			Tono		
				C	B	A	C	B	A
8	Sonoridad canal A				M	L (A)			
9	Sonoridad canal B				M	L (B)			
10	Sonoridad canal C				M	L (C)			
11	Periodo de la envolvente	FT (E)							
12		CT (E)							
13	Pauta de la envolvente					EP			

Fig. 2.11. Tabla de funciones de los registros de sonido en MSX.

nos significativos corresponden al primer registro y los 4 bits más significativos al segundo. Para determinar el tono podemos utilizar la fórmula:

$$T = 1789772.5/(16*F)$$

donde *F* es la frecuencia correspondiente a la nota deseada (podemos consultar los valores de las frecuencias en la tabla de la figura 2.4). Una vez determinado el valor de T podemos calcular los 4 bits más significativos mediante la fórmula:

$$BS = T/256$$

Los 8 bits menos significativos se obtienen con la fórmula:

$$BM = T \text{ MODE } 256$$

El valor de BS puede estar comprendido entre 0 y 15, ya que los valores superiores contienen más de cuatro bits, mientras que el valor de BM puede variar entre 0 y 255. Los valores más pequeños producen tonos más agudos, mientras que si aumentamos los valores los tonos se irán haciendo más graves.

Recordemos ahora que el valor de BM debe almacenarse en los registros 0, 2 ó 4 y el de BS en los impares (1, 3 ó 5).

Para seleccionar el canal de salida tenemos que introducir en el regis-

tro 7 un número en notación binaria o su decimal correspondiente. La selección del canal se realiza colocando un cero en el bit correspondiente al canal deseado, teniendo en cuenta que, considerando los bits de derecha a izquierda, el primer bit da salida de tono al canal A, el segundo al B y el tercero al C; el cuarto bit da salida de ruido al canal A, el quinto al B y el sexto al C. Los dos últimos bits siempre valen 0 y 1 y están al margen de la función de salida de sonido. Por ejemplo la instrucción:

SOUND 7,129

selecciona el canal A para salida de tono ya que 129=10000001. También podríamos introducirlo en notación binaria de la forma:

SOUND 7,&B10000001

En la figura 2.12 podemos ver un esquema de selección del canal de salida.

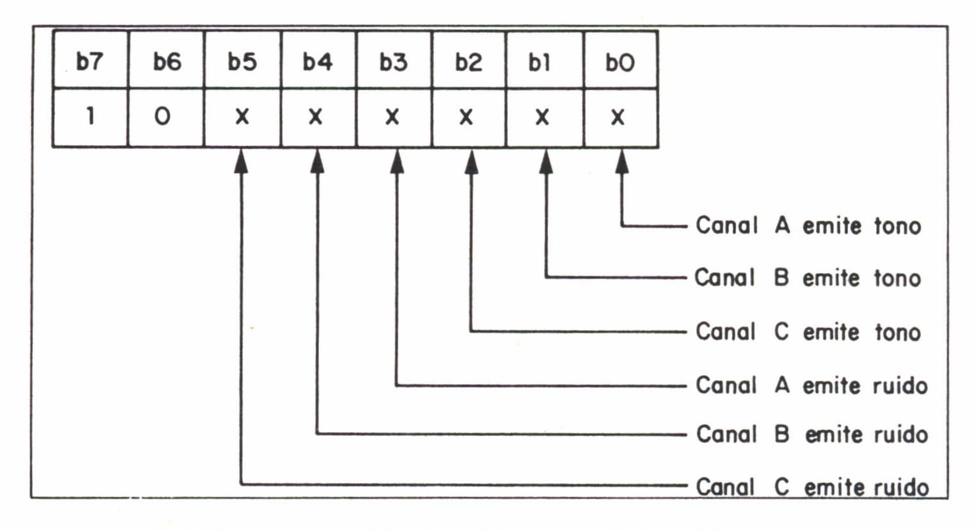


Fig. 2.12. Esquema de selección del canal de salida en MSX.

El nivel de volumen de los canales A, B y C se ajusta en los registros 8, 9 y 10 respectivamente. El quinto bit (desde la derecha) de cada registro indica si el sonido se ajusta a la envolvente definida con un 1, o si el nivel es constante con 0.

Los primeros cuatro bits determinan el volumen con valores de 0 (mínimo) a 15 (máximo). Por ejemplo, la instrucción:

SOUND 9,26

ajusta la sonoridad del canal B a la envolvente y volumen 10, ya que 26=00011010. Recordemos que para la conversión de un número binario

a notación decimal debemos seguir el esquema representado en la figura 2.13.

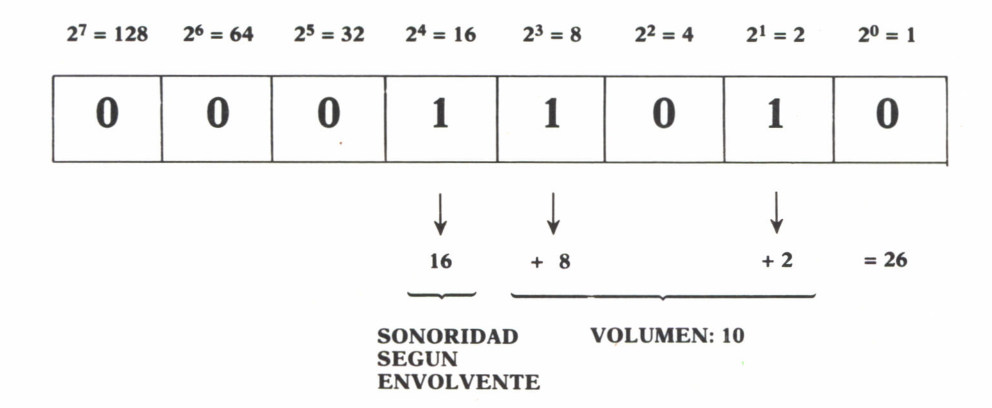


Fig. 2.13. Esquema de conversión binario-decimal para la selección de la intensidad sonora en MSX.

El período de la envolvente indica la relación tiempo-intensidad sonora. Se indica en los registros 11 y 12 con valores comprendidos entre 0 y 255 en cada registro. Estos valores se obtienen con las fórmulas:

PT = 1789772.5*T/256
BT = PT/256 (8 bits más significativos)
BM = PT MODE 256 (8 bits menos significativos)

El valor obtenido en BT se carga en el registro 11 y el de BM en el registro 12.

Finalmente, el registro 13 nos permite seleccionar la forma de la onda sonora a la que queremos ajustar nuestro sonido. Podemos introducir un número entre 0 y 15 para seleccionar cualquiera de los ocho tipos de onda existentes, tal y como muestra la figura 2.14.

Para terminar, el programa 2.14 produce un interesante sonido de campanas.

```
10 REM **********************
20 REM * SONIDO DE CAMPANAS *
30 REM *        MSX         *
40 REM **********************
50 CLS:PRINT TAB(16);"CAMPANAS"
```

```
60 FOR R=0 TO 13
70 READ V
80 SOUND R,V
90 NEXT
100 RESTORE
110 GOTO 60
120 DATA 70,1,70,1,0,2,0
```

Programa 2.14.

El bucle FOR-NEXT de las líneas 60-90 se encarga de leer los valores a introducir en cada registro, que están almacenados en líneas DATA, y a continuación los carga con la instrucción SOUND.

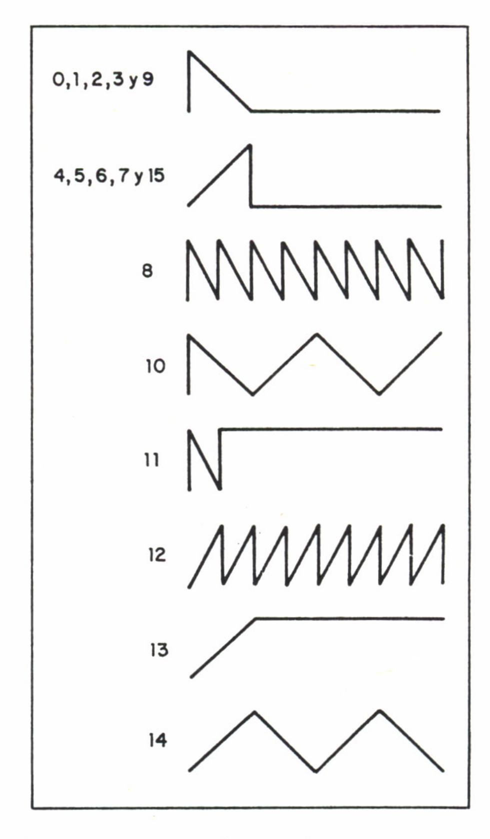
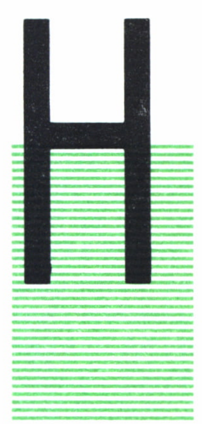


Fig. 2.14. Tipos de envolventes en MSX.

LAS PRIMERAS MELODIAS 3



HASTA ahora hemos visto cómo podemos generar sonidos en distintos ordenadores, e incluso, ya hemos conseguido que el ordenador ejecute diversas escalas, naturales y cromáticas, ascendentes y descendentes.

De modo que ya va siendo hora de pasar a interpretar las primeras melodías. Vamos a comenzar por trasladar a las instrucciones BASIC vistas una serie de melodías hasta cierto punto sencillas. Bien es verdad que la melodía más sencilla que podemos encontrar es aquella en la que la duración de todas sus notas es la misma, siendo el tono la única característica variable. Desde luego este tipo de melodías son bastante escasas y siempre suelen resultar algo «sosas». En cualquier caso nosotros ya contamos con suficientes conocimientos para permitirnos programar melodías con, al menos, dos parámetros variables: tono y duración.

El método a seguir es el mismo en todos los ordenadores. Almacenaremos en líneas DATA toda la información referente a cada nota, en principio tono y duración, y la melodía se ejecuta mediante un bucle que lee dicha información e interpreta cada nota con la instrucción de sonido adecuada. De todos modos vamos a dividir el capítulo de igual forma que el anterior para ir viendo diferentes ejemplos en distintos ordenadores de una forma cómoda y clara.



LAS MELODIAS DEL SPECTRUM

Ya sabemos que en el SPECTRUM el tono y la duración constituyen la máxima información posible que podemos establecer para la generación de sonido, por tanto, lo único que puede variar es el grado de complejidad de una melodía en sí misma.

Vamos a comenzar por estudiar el conocido villancico del programa 3.1.

```
10 REM ******************
20 REM * ADESTE FIDELES *
30 REM *    SPECTRUM    *
40 REM ******************
50 PRINT AT 8,9;"ADESTE FIDELES"
60 PRINT AT 11,11;"VILLANCICO"
70 DIM F(8):DIM N$(8,3)
80 FOR I=1 TO 8
90 READ F(I),N$(I)
100 NEXT I
110 DATA 2,"RE",4,"MI",6,"FA#",7,"SOL"
120 DATA 9,"LA",11,"SI",12,"DO",14,"RE"
130 FOR I=1 TO 61
140 READ D,N
150 PRINT AT 11,15;N$(N);" "
160 BEEP D,F(N)
170 NEXT I
180 DATA 0.5,4,1,4,0.5,1,0.5,4
190 DATA 1,5,1,1,0.5,6,0.5,5,0.5,6,0.5,7
200 DATA 1,6,0.5,5,0.5,4,1,4,0.5,3,0.5,2
210 DATA 0.5,3,0.5,4,0.5,5,0.5,6,1,3
    ,0.75,2,0.25,1
220 DATA 2,1,1,8,0.5,7,0.5,6
230 DATA 1,7,1,6,0.5,5,0.5,6,0.5,4,0.5,5
240 DATA 1,3,0.5,1,0.5,4,0.5,4,0.5,3,0.5
    ,4,0.5,5
250 DATA 1,4,0.5,1,0.5,6,0.5,6,0.5,5,0.5
    ,6,0.5,7
260 DATA 1,6,0.5,5,0.5,6,0.5,7,0.5,6,0.5
    ,5,0.5,4
270 DATA 1,3,0.5,4,0.5,7,1,6,0.75,5,0.25
    ,4,1,4
```

Programa 3.1.

En este programa hemos utilizado un truco que resulta bastante cómodo si no queremos estar manejando frecuencias continuamente. Si nos fijamos, toda la melodía se interpreta únicamente con ocho tonos distintos (una octava de RE a RE). Lo que hacemos es asignar a estas ocho notas ocho números consecutivos, pero debe quedar claro que este sistema de numeración lo elegimos arbitrariamente y no tiene ninguna relación con el tono de cada nota. La figura 3.1 aclara este punto.

Frecuencia	2	4	6	7	9	11	12	14
Nota	**RE**	**MI**	**FA**	**SOL**	**LA**	**SI**	**DO**	**RE**
N.º Asignado	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Fig. 3.1. Relación entre frecuencia, nota y número asignado en el villancico «Adeste Fideles» (SPECTRUM).

De este modo podemos crear dos matrices de ocho elementos cada una (línea 70). La matriz numérica almacena las frecuencias de cada una de las ocho notas, mientras que la matriz alfanumérica almacena los nombres de dichas notas. Para ello sólo es necesario realizar una única lectura de cada uno de estos datos almacenados en las líneas DATA 110 y 120. La lectura se realiza con el bucle FOR-NEXT de las líneas 80-100 que se repite ocho veces, cargando simultáneamente las dos matrices. Ahora ya podemos olvidarnos de las frecuencias y los nombres de las notas. En las líneas DATA de la 180 a la 270 sólo almacenamos la duración de cada nota y el número que le hemos asignado. El bucle FOR-NEXT de las líneas 130-170 se encarga de leer estos datos 61 veces, ya que la melodía se compone de 61 notas. Cada vez que se ejecuta el bucle se imprime en pantalla el nombre de la nota que va a sonar (línea 150) y se genera el sonido adecuado (línea 160).

Veamos ahora otro ejemplo un poco más complicado. Vamos a trasladar la partitura de la figura 3.2 a un programa BASIC en el SPECTRUM.

Podemos observar que en esta conocida melodía hay un tema principal que se repite varias veces, incluso en distintas octavas, por lo tanto no es necesario trasladar a líneas DATA todas y cada una de las notas, sino sólo el tema principal y las partes de la melodía que se salgan de este tema. Luego bastará con mandar el puntero de datos a la línea DATA adecuada mediante la instrucción RESTORE y conseguiremos escuchar la melodía completa utilizando una serie de bucles FOR-NEXT. Todo esto está desarrollado en el programa 3.2.

En este caso hemos elegido la escala que va de SOL (frecuencia −5) a FA (frecuencia 5) y hemos numerado las notas de 1 a 7. El primer bucle FOR-NEXT (líneas 80-100) se utiliza para leer las frecuencias y los nombres de estas siete notas y almacenarlos en las matrices correspondientes. El siguiente bucle (líneas 130-150) interpreta los siete primeros compases de la partitura. A continuación se repite de nuevo el tema principal por lo que mandamos el puntero de datos a la primera línea DATA con la instruc-
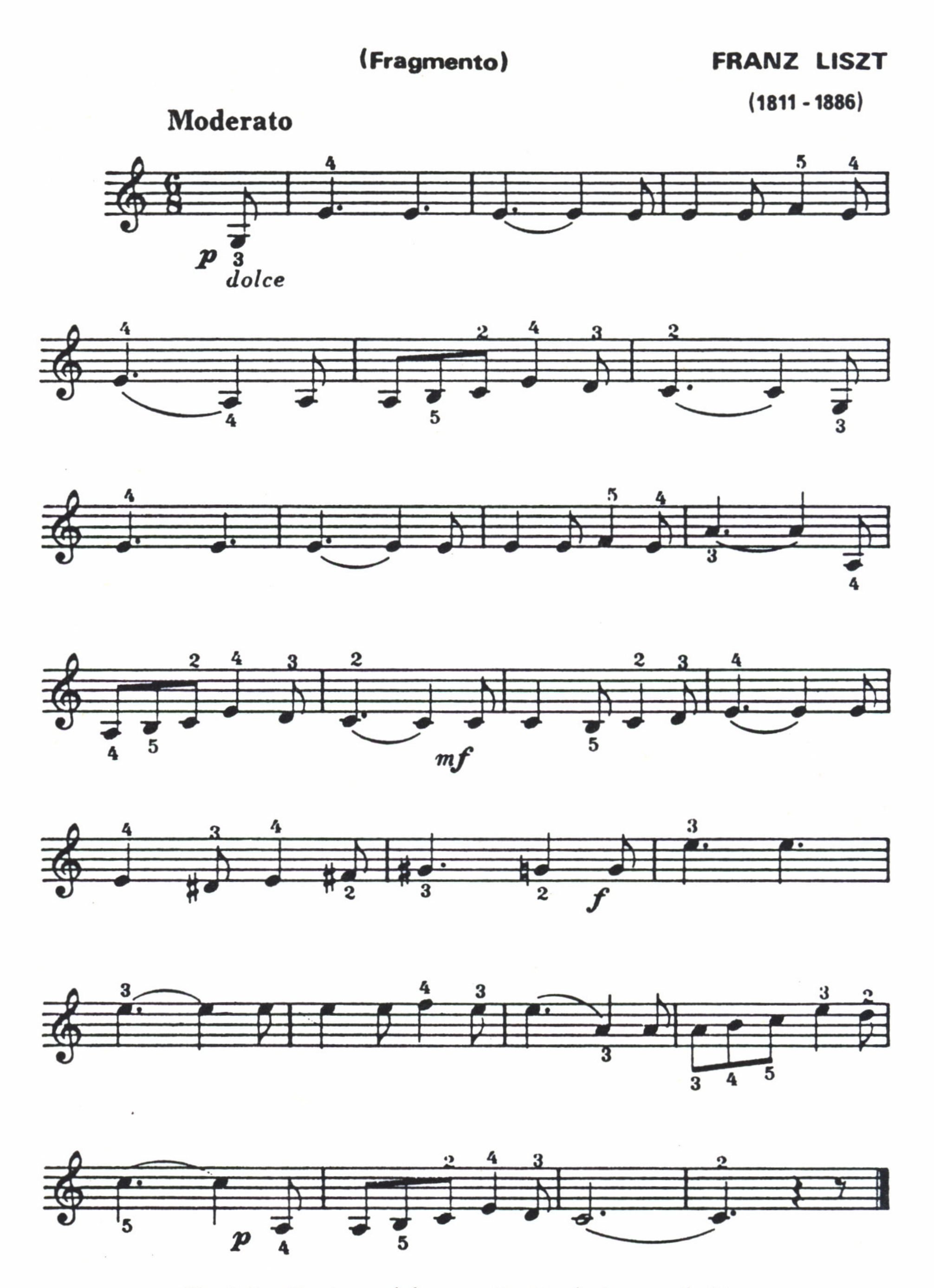


Fig. 3.2. Partitura del tema «Sueño de Amor», de Liszt.

```
10 REM ***************************
20 REM * SUEÑO DE AMOR (LISZT) *
30 REM *       SPECTRUM         *
40 REM ***************************
50 PRINT AT 8,9;"SUEÑO DE AMOR"
60 PRINT AT 11,10;"FRANZ LISZT"
70 DIM F(7),DIM N$(7,3)
80 FOR I=1 TO 7
90 READ F(I),N$(I)
100 NEXT I
110 DATA -5,"SOL",-3,"LA",-1,"SI"
120 DATA 0,"DO",2,"RE",4,"MI",5,"FA"
130 FOR I=1 TO 18
140 GOSUB 500
150 NEXT I
160 RESTORE 600
170 FOR I=1 TO 9
180 GOSUB 500
190 NEXT I
200 PRINT AT 11,15;N$(2):BEEP 1,9
210 RESTORE 620
220 FOR I=1 TO 15
230 GOSUB 500
240 NEXT I
250 RESTORE 640
260 FOR I=1 TO 5
270 READ P$,D,F
280 PRINT AT 15,15;P$;" "
290 BEEP D,F
300 NEXT I
310 RESTORE 600
320 for I=1 TO 18
330 READ D,N
340 PRINT AT 15,15;N$(N)
350 BEEP D,F(N)+12
360 NEXT I
370 RESTORE 620
380 FOR I=1 TO 7
390 READ D,N
400 PRINT AT 15,15;N$(N)
410 BEEP D,F(N)+2
420 NEXT I
430 RESTORE 620
440 FOR I=1 TO 7
450 GOSUB 500
460 NEXT I
470 GO TO 9000
500 READ D,N
510 PRINT AT 15,15;N$(N)
```

```
520 BEEP D,F(N)
530 RETURN
600 DATA 0.3,1,0.9,6,0.9,6,1.5,6,0.3,6
    ,0.6,6,0.3,6,0.6,7,0.3,6
610 DATA 0.9,6,0.6,2
620 DATA 0.3,2,0.3,2,0.3,3,0.3,4,0.6,6
    ,0.3,5,1.5,4
630 DATA 0.3,4,0.6,4,0.3,3,0.6,4,0.3,5
    ,1.5,6,0.3,6,0.6,6
640 DATA "RE#",0.3,3,"MI",0.6,4,"FA#"
    ,0.3,6,"SOL#",0.9,8,"SOL",0.6,7
```

Programa 3.2.

ción **RESTORE** de la línea 160. Este esquema de ir seleccionando la línea **DATA** adecuada se repite a lo largo de todo el programa.



LAS MELODIAS DEL AMSTRAD

Recordemos que el **AMSTRAD** puede utilizar hasta tres canales de sonido simultáneamente, pudiéndose definir para cada uno hasta siete parámetros de sonido. Sin embargo, para comenzar con las primeras interpretaciones musicales vamos a utilizar un único canal, el **A**, y sólo vamos a variar dos parámetros: el tono y la duración. En capítulos posteriores alcanzaremos mayor complejidad al utilizar todos los parámetros y varios canales de sonido simultáneamente.

Vamos a comenzar por uno de esos raros ejemplos en el que todas las notas que componen la melodía (excepto tres) tienen la misma duración. El programa 3.3 interpreta en el **AMSTRAD** el popular «Romance Anónimo».

```
10 REM *******************
20 REM * ROMANCE ANONIMO *
30 REM *     AMSTRAD     *
40 REM *******************
50 CLS
60 LOCATE 13,13:PRINT "ROMANCE ANONIMO"
70 DIM T(12):DIM N$(12)
80 FOR I=0 TO 12
90 READ T(I),N$(I)
```

```
100 NEXT
110 DATA 239,DO,225,DO#,213,RE,190,MI,179,FA
120 DATA 169,FA#,159,SOL,142,LA,134,LA#
130 DATA 127,SI,119,DO,113,DO#,106,RE
140 FOR I=1 TO 42
150 GOSUB 390
160 NEXT
170 DATA 7,7,7,7,6,4,4,3,2
180 DATA 2,4,7,12,12,12,12,10,8
190 DATA 8,7,6,6,7,8,7,8,7
200 DATA 10,8,7,7,6,4,4,3,2
210 DATA 3,3,3,3,4,3
220 LOCATE 20,16:PRINT N$(2);"   "
230 SOUND 1,T(2),120,15
240 WHILE SQ(1)>128:WEND
250 FOR I=1 TO 41
260 GOSUB 390
270 NEXT
280 DATA 5,5,5,5,3,2,2,1,1
290 DATA 1,0,1,9,9,9,9,11,9
300 DATA 9,7,7,7,9,11,12,12,12
310 DATA 12,11,10,9,9,9,9,7,6
320 DATA 5,5,5,5,6
330 LOCATE 20,16:PRINT N$(3);"   "
340 SOUND 1,T(3),60,15
350 WHILE SQ(1)>128:WEND
360 LOCATE 20,16:PRINT N$(2);"   "
370 SOUND 1,T(2),120,15
380 END
390 READ N
400 LOCATE 20,16:PRINT N$(N);"   "
410 SOUND 1,T(N),40,15
420 WHILE SQ(1)>128:WEND
430 RETURN
```

Programa 3.3.

La melodía se compone combinando 13 tonos diferentes, que están almacenados, junto con los nombres de las notas correspondientes, en las líneas **DATA** de la 110 a la 130. Estas trece parejas de datos las almacenamos en dos matrices, una de tonos y otra de nombres de notas de modo que el primer elemento de ambas matrices (el 0) corresponde a la nota más grave (el **DO**, tono 239), mientras que el último elemento de la matriz (el 12) corresponde a la nota más aguda (el **RE**, tono 106), tal y como muestra la figura 3.3.

Tono	239	225	213	190	179	169	159	142	134	127	119	113	106
Nota	**DO**	**DO#**	**RE**	**MI**	**FA**	**FA#**	**SOL**	**LA**	**LA#**	**SI**	**DO**	**DO#**	**RE**
Elemento de la matriz	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**

Fig. 3.3. Relación entre tono, nota y elemento de la matriz en el tema «Romance Anónimo» (AMSTRAD).

Ambas matrices se cargan con el bucle FOR-NEXT de las líneas 80-100. Seguidamente la melodía se divide en dos partes. La primera se interpreta con el bucle de las líneas 140-160 y la segunda con el bucle de las líneas 250-270.

El funcionamiento de ambos bucles es el mismo. Cada vez que dan una vuelta, el control del programa se transfiere a la subrutina 390 donde se realiza la lectura de la nota que va a sonar (línea 390), representada en las líneas DATA por su posición en las matrices. A continuación se imprime el nombre de la nota en la pantalla (línea 400) y seguidamente se produce el sonido (línea 410). El volumen se ha ajustado al máximo (15) y la duración se ha establecido en 40 centésimas.

Finalmente encontramos una novedad en la línea 420 (que también se repite en otras líneas a lo largo del programa). Un bucle WHILE-WEND que utiliza una función SQ, hasta ahora desconocida. Veamos su misión.

El AMSTRAD tiene la capacidad de generar «colas» de sonido, es decir, se pueden almacenar varios sonidos en una memoria intermedia y se irán ejecutando a medida que llega su turno, sin detener la ejecución del programa. La función SQ tiene por objeto informar de la situación de la cola de sonido del canal especificado. SQ tiene siempre el formato:

SQ (canal)

donde *canal* será 1 para el canal A, 2 para el B y 4 para el C.

La función SQ produce un número que se interpreta, según sea el valor de los bits, de la forma siguiente:

Bits 0, 1 y 2: número de huecos libres en la cola.
Bits 3, 4 y 5: estado de sincronización de la primera nota de la cola.
Bit 6: la primera nota está retenida en la cola.
Bit 8: el canal está activo.

Por lo tanto una línea de tipo:

WHILE SQ(1) > 128 : WEND

como las que aparecen en el programa 3.3, tiene por objeto generar una pequeña pausa entre una nota y la siguiente de modo que éstas no se acumulan en la cola de sonido. De este modo la interpretación de la melodía y la ejecución del programa van a la par.

Si no estableciéramos estas pausas en el programa 3.3, las notas que se imprimen en pantalla irían a mayor velocidad que la melodía por lo que no habría una correspondencia. Además el mensaje «Ready» que indica el final de la ejecución del programa aparecería mucho antes de que finalizara la melodía ya que ésta estaría almacenada en la «cola» de sonido.

A continuación vamos a trasladar al AMSTRAD la partitura representada en la figura 3.4.

En esta melodía aumenta un poco el nivel de complejidad ya que la duración de las notas es variable. Lo que hacemos en el programa 3.4 es almacenar en líneas DATA parejas de números, correspondientes al tono y la duración de cada nota. Para la interpretación de la pieza la dividimos en tres partes ya que hay unos cuantos compases que se repiten (del 10 al 14).

Estas tres partes se hacen corresponder con tres bucles FOR-NEXT que interpretan respectivamente la primera parte, el tema que se repite, y la parte final. El esquema para interpretar cada nota es prácticamente el mismo que en el programa anterior: una subrutina (línea 200) que realiza la lectura de los datos, genera el sonido y produce una pausa con un bucle WHILE-WEND y la función SQ.



LAS MELODIAS DEL IBM

Como ya sabemos el IBM puede interpretar melodías con dos instrucciones diferentes: SOUND y PLAY. En este capítulo vamos a ver dos ejemplos, cada uno con una instrucción, de las posibilidades musicales del IBM.

Recordemos que SOUND tiene únicamente dos parámetros, al igual que el BEEP del SPECTRUM, por tanto, podemos observar que el desarrollo del programa 3.5 es muy similar a los programas vistos anteriormente para el SPECTRUM (programas 3.1 y 3.2).

Por otra parte la instrucción PLAY nos permite interpretar cadenas de mandatos musicales lo que da lugar a programas más cortos. En la figura 3.5 podemos ver la partitura de un fragmento del conocido «Vals de las Olas».

Al igual que en ejemplos anteriores hay un tema que se repite. En este caso dicho tema lo vamos a asignar a la variable T$ de modo que cada vez que haya que interpretarlo no tendremos que llamar a dicha variable desde la instrucción PLAY correspondiente. El resto de la melodía lo tradu-
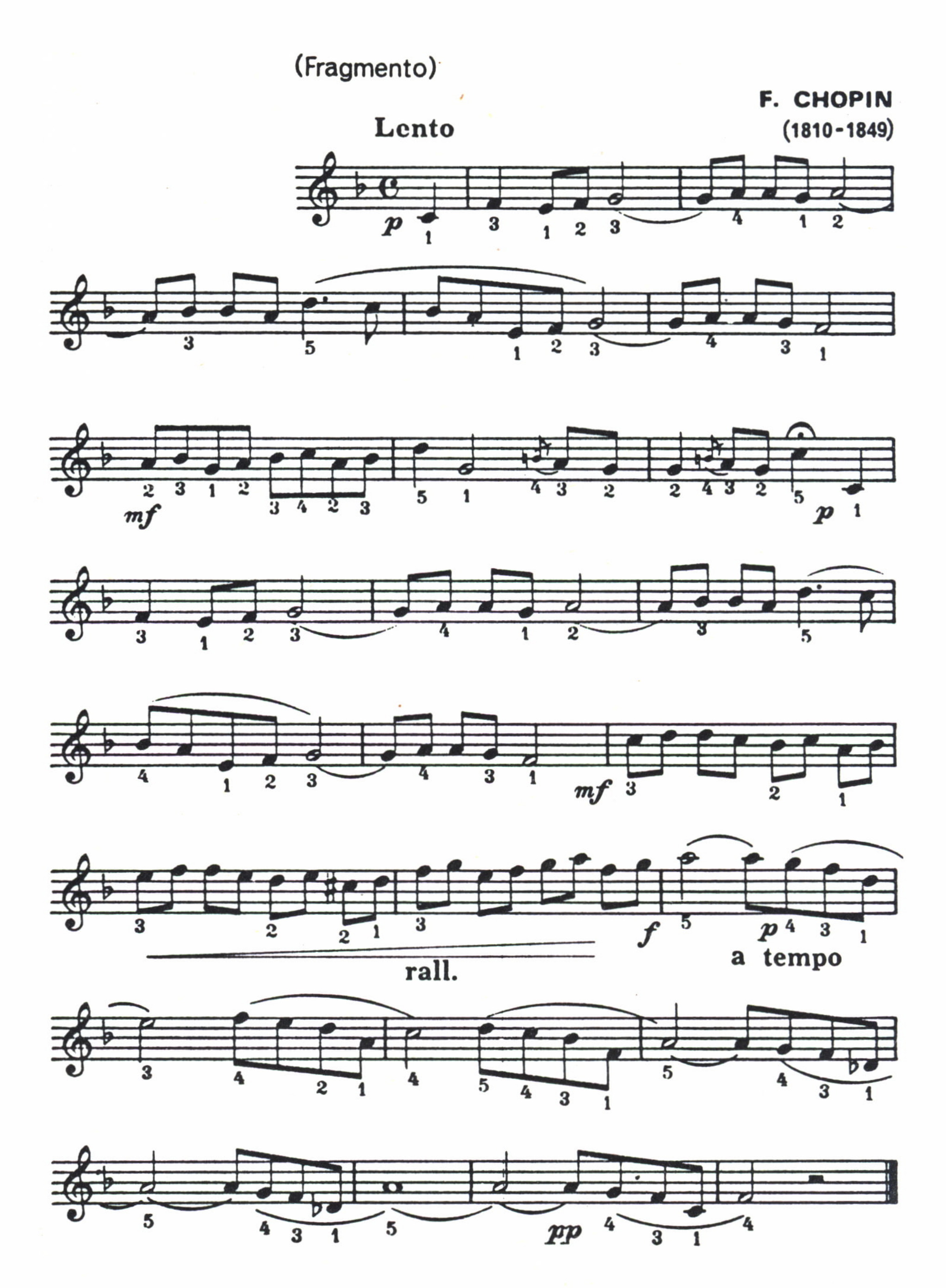


Fig. 3.4. Partitura del tema «Tristeza», de Chopin.

```
10 REM *********************
20 REM * TRISTEZA (CHOPIN) *
30 REM *      AMSTRAD      *
40 REM *********************
50 CLS
60 LOCATE 17,10:PRINT "TRISTEZA"
70 LOCATE 13,13:PRINT "FEDERICO CHOPIN"
80 FOR I=1 TO 39
90 GOSUB 200
100 NEXT
110 RESTORE
120 FOR I=1 TO 23
130 GOSUB 200
140 NEXT
150 RESTORE 320
160 FOR I=1 TO 51
170 GOSUB 200
180 NEXT
190 END
200 READ T,D
210 SOUND 1,T,D,15
220 WHILE SQ(1)>128:WEND
230 RETURN
240 DATA 239,80,179,80,190,40,179,40,159,200
250 DATA 142,40,142,40,159,40,142,200
260 DATA 134,40,134,40,142,40,106,120,119,40
270 DATA 134,40,142,40,190,40,179,40,159,200
280 DATA 142,40,142,40,159,40,179,160
290 DATA 142,40,134,40,159,40,142,40,134,40
295 DATA 119,40,142,40,134,40
300 DATA 106,80,159,160,142,40,159,40
310 DATA 159,80,142,40,159,40,119,80
320 DATA 119,40,106,40,106,40,119,40,134,40
325 DATA 119,40,142,40,134,40
330 DATA 95,40,89,40,89,40,95,40,106,40,95
335 DATA 40,113,40,106,40
340 DATA 89,40,80,40,95,40,89,40,80,40,71
345 DATA 40,89,40,80,40
350 DATA 71,200,80,40,89,40,106,40
360 DATA 95,160,89,40,95,40,106,40,142,40
370 DATA 119,160,106,40,119,40,134,40,179,40
380 DATA 142,200,159,40,179,40,225,40
390 DATA 142,200,159,40,179,40,225,40
400 DATA 142,520,159,40,179,40,239,40,179,160
```

Programa 3.4.

cimos a mandatos musicales de **PLAY** vigilando sobre todo las octavas que corresponden y las duraciones de las notas. (Recordemos que L4 es una

```
10 REM ***************
20 REM * SANTA LUCIA *
30 REM *     IBM     *
40 REM ***************
50 KEY OFF:WIDTH 40:CLS
60 LOCATE 10,15:PRINT "SANTA LUCIA"
70 DIM F(10),N$(10)
80 FOR I=1 TO 10
90 READ F(I),N$(I)
100 NEXT
110 DATA 293.66,RE,329.63,MI,349.23,FA,
369.99,FA#
120 DATA 391.99,SOL,440,LA,493.88,SI
130 DATA 523.25,DO,587.33,RE,659.25,MI
140 FOR I=1 TO 50
150 READ D,N
160 LOCATE 15,20:PRINT N$(N);"  "
170 SOUND F(N),D*1.5:SOUND 32767,1
180 NEXT
190 DATA 4,5,6,5,2,8,3,8,1,7,8,7
200 DATA 4,3,6,3,2,6,3,6,1,5,8,5
210 DATA 4,2,6,6,2,5,3,5,1,4,8,3
220 DATA 4,3,4,2,4,1,3,6,1,5,8,5
230 DATA 4,10,4,9,4,8,3,7,1,6,8,9
240 DATA 4,9,4,8,4,6,3,4,1,5,8,8
250 DATA 2,10,2,8,2,8,2,5,2,5,2,2,4,3,8,9
260 DATA 4,9,6,10,2,9,3,9,1,8,8,8
```

Programa 3.5.

negra y L1 es una redonda o, como en este caso, una blanca con puntillo ligada a una negra). Todo esto está recogido en el programa 3.6.



LAS MELODIAS DEL COMMODORE

Finalmente vamos a ver cómo programar bonitas melodías en el COMMODORE que, aunque no dispone de comandos de sonido, es tan fácil y espectacular como en los demás ordenadores.

Vamos a comenzar por el programa 3.7 que reproduce un fragmento del conocido «Minueto», de Beethoven.
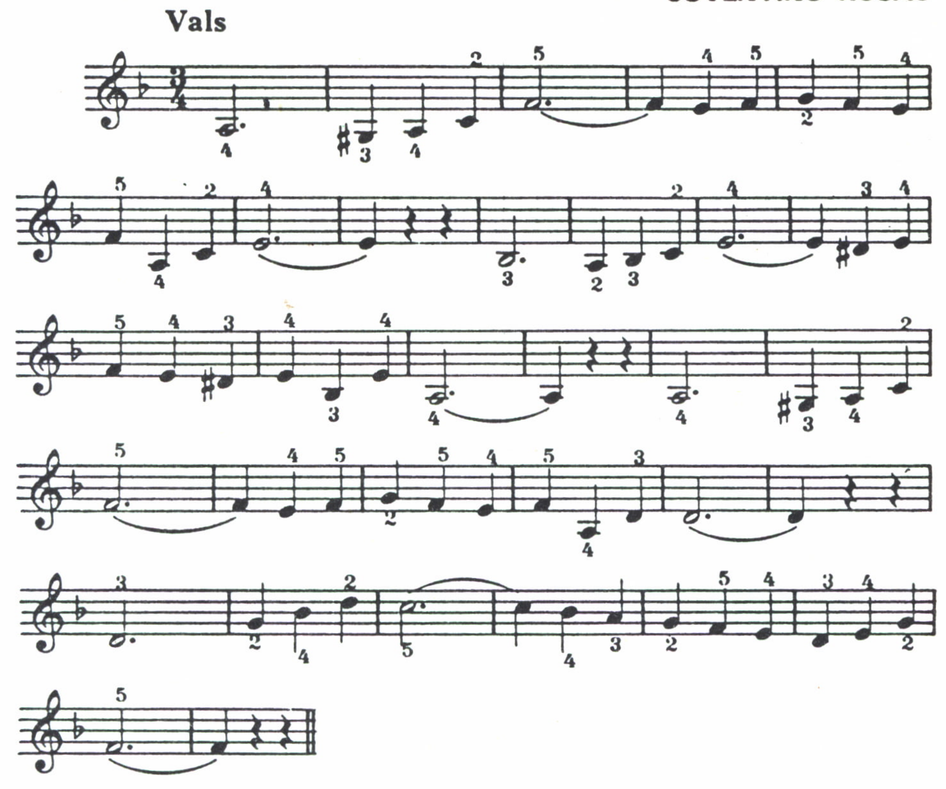


Fig. 3.5. Partitura del «Vals de las Olas», de Rosas.

```
10 REM ******************************
20 REM * VALS DE LAS OLAS (ROSAS) *
30 REM *            IBM              *
40 REM ******************************
50  KEY OFF:WIDTH 40:CLS
60 LOCATE 10,12:PRINT "VALS DE LAS OLAS"
70 LOCATE 14,12:PRINT "JUVENTINO ROSAS"
80 T$="T200 O2 A2. L4 G# A O3 C F1 L4 E
F G F E F O2 A"
90 PLAY "XT$; O3 C E1 P2"
100 PLAY "O2 B-2. L4 A B- O3 C E1 D# E F
E D# E O2 B- O3 E O2 A1 P2"
110 PLAY "XT$; O3 D D1 P2"
120 PLAY "O3 D2. L4 G B- O4 D C1 O3 B- A
G F E D E G F1 P2"
```

Programa 3.6.

```
10 REM ************
20 REM *  MINUETO  *
30 REM * BEETHOVEN *
40 REM ************
50 PRINT CHR$(147)
60 PRINT SPC(216);"MINUETO"
70 PRINT SPC(210);"LUDWIG VAN BEETHOVEN"
80 FOR I=54272 TO 54296
90 POKE I,0
100 NEXT I
110 DIM A(8),B(8)
120 FOR I=1 TO 8
130 READ A(I),B(I)
140 NEXT I
150 DATA 24,63,25,177,28,214,32,94
160 DATA 34,75,36,85,38,126,43,52
170 POKE 54296,15
180 POKE 54278,240
190 FOR I=1 TO 27
200 POKE 54276,17
210 READ D,N
220 POKE 54272,B(N)
230 POKE 54273,A(N)
240 FOR T=1 TO D:NEXT T
250 POKE 54276,16
260 NEXT I
270 DATA 240,2,80,1,640,1,240,1,80,2,640,2
280 DATA 240,2,80,4,480,7,160,8,160,7,160,5
290 DATA 320,4,320,2
300 DATA 240,2,80,1,640,1,240,1,80,2,640,2
310 DATA 240,2,80,4,480,7,160,6,160,5,160,3
320 DATA 320,2
```

Programa 3.7.

Veamos detenidamente el funcionamiento del programa. El primer bucle **FOR-NEXT** (líneas 80-100) se encarga, como ya hemos dicho en el capítulo anterior, de poner a cero todos los registros de sonido. A continuación se dimensionan dos matrices numéricas de ocho elementos cada una (línea 110) que tienen por objeto almacenar los datos sobre altas y bajas frecuencias de las ocho notas utilizadas a lo largo de la melodía. Además también establecemos el tempo de la melodía que determinará su velocidad de interpretación.

El bucle de las líneas 120-140 carga las dos matrices con los datos al-

macenados en las líneas DATA 150 y 160. A continuación ajustamos el volúmen al máximo (línea 170) y definimos el nivel de ataque-caída y sostenimiento-relajación (línea 180). La interpretación de la melodía se realiza con el bucle FOR-NEXT de las líneas 190-260, dentro del cual se ajusta la forma de la onda (línea 200), se leen los datos correspondientes a la duración y nota (línea 210) y se introducen los valores adecuados de alta y baja frecuencia en los registros correspondientes del canal A (líneas 220 y 230). Finalmente en las líneas DATA se almacenan parejas de datos que definen la duración y la nota.

El programa 3.8 es otro ejemplo de las posibilidades sonoras del COMMODORE. En este caso interpreta la famosa «Canción de Cuna» de Brahms.

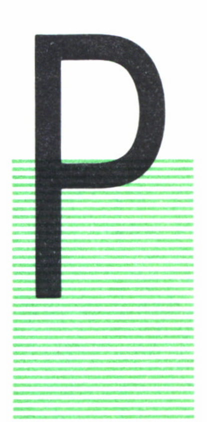
```
10 REM *****************************
20 REM * CANCION DE CUNA (BRAHMS) *
30 REM *        COMMODORE          *
40 REM *****************************
50 PRINT CHR$(147)
60 PRINT SPC(212);"CANCION DE CUNA"
70 PRINT SPC(217);"BRAHMS"
80 SID=54272
90 FOR I=SID TO SID+24
100 POKE I,0
110 NEXT I
120 DIM A(8),B(8)
130 FOR I=1 TO 8
140 READ A(I),B(I)
150 NEXT I
160 POKE SID+24,15
170 POKE SID+6,240
180 POKE SID+4,17
190 READ D
200 IF D<0 THEN END
210 READ N
220 POKE SID,B(N)
230 POKE SID+1, A(N)
240 FOR I=1 TO D*160:NEXT I
250 POKE SID+4,16
260 GOTO 180
270 DATA 22,227,25,177,28,214,30,141
280 DATA 34,75,38,126,43,52,45,198
290 DATA 1,3,1,3
300 DATA 4,5,1,3,1,3
310 DATA 4,5,1,3,1,5
320 DATA 2,8,2,7,2,6
330 DATA 2,6,2,5,1,2,1,3
```

```
340 DATA 4,4,1,2,1,3
350 DATA 4,4,1,2,1,4
360 DATA 1,7,1,6,2,5,2,7
370 DATA 4,8,1,1,1,1
380 DATA 2,8,2,6,2,4
390 DATA 4,5,1,3,1,1
400 DATA 2,4,2,5,2,6
410 DATA 2,3,2,5,1,1,1,1
420 DATA 2,8,2,6,2,4
430 DATA 4,5,1,3,1,1
440 DATA 0.67,4,0.67,5,0.67,4,2,3,2,2
450 DATA 4,1,-1
```

Programa 3.8.

El esquema del programa es el mismo que en el ejemplo anterior. En este caso hemos establecido una variable SID=54272 para identificar el primer registro de sonido. Los demás registros están establecidos en función de esta variable SID (por ejemplo, SID + 24 = 54296, registro del volumen). La otra novedad que encontramos en el programa es que aquí no hay un bucle FOR-NEXT para interpretar la melodía sino que al final de los DATA establecemos un dato ficticio, -1, que indica el final de la melodía, de modo que cuando se produce la lectura de ese dato, mediante la condición adecuada finaliza la ejecución del programa.

MUSICA, MAESTRO 4



POR fin ha llegado el momento de utilizar el ordenador para interpretaciones más sofisticadas. Vamos a dedicar el presente capítulo al desarrollo de un tema musical completo en el SPECTRUM, para pasar a continuación al tema estrella del sonido con ordenador: la interpretación de melodías con varias voces utilizando simultáneamente varios canales de sonido. Finalmente veremos un interesante programa musical para IBM que nos permite seleccionar la melodía que queremos escuchar.



EL SPECTRUM Y BEETHOVEN

Si Beethoven levantara la cabeza... quedaría totalmente perplejo al comprobar que un «aparatito» como el SPECTRUM es capaz de interpretar su célebre obra «Para Elisa» completa, desde la primera hasta la última nota, y sin equivocarse.

El programa 4.1 hace posible este «milagro» de la técnica.

```
 1 REM ************************
 2 REM *PARA ELISA(BEETHOVEN)*
 3 REM *        SPECTRUM     *
 4 REM ************************
 6 CLS
 7 PRINT AT 9,11;"PARA ELISA"
 8 PRINT AT 13,11;"BEETHEVEN"
10 FOR i=1 TO 58
20 READ a,b
30 BEEP a,b
```

```
 40 NEXT i
 50 DATA .2,17,.2,16
 60 DATA .2,17,.2,16,.2,17,.2,12,.2,15,.2,13
 70 DATA .6,10,.2,1,.2,5,.2,10
 80 DATA .6,12,.2,5,.2,9,.2,12
 90 DATA .6,13,.2,5,.2,17,.2,16
100 DATA .2,17,.2,16,.2,17,.2,12,.2,15,.2,13
110 DATA .6,10,.2,1,.2,5,.2,10
120 DATA .6,12,.2,5,.2,13,.2,12
130 DATA .6,10
140 DATA .2,12,.2,13,.2,15
150 DATA .6,17,.2,8,.2,18,.2,17
160 DATA .6,15,.2,6,.2,17,.2,15
170 DATA .6,13,.2,5,.2,15,.2,13
180 DATA .6,12,.2,5,.6,17
190 DATA .2,17,.6,29,.2,16
200 DATA .6,17,.2,16
210 RESTORE 50
220 FOR i=1 TO 35
230 READ a,b
240 BEEP a,b
250 NEXT i
260 RESTORE 310
270 FOR i=1 TO 102
280 READ a,b
290 BEEP a,b
300 NEXT i
310 DATA .2,13,.2,13,.2,13
320 DATA .8,13,.3,18,.1,17
330 DATA .4,17,.4,15,.3,23,.1,22
340 DATA .2,22,.2,20,.2,18,.2,17,.2,15,.2,13
350 DATA .4,11,.4,10,.1,10,.1,8,.1,10,.1,11
360 DATA .8,13,.2,15,.2,16
370 DATA .6,17,.2,17,.2,18,.2,10
380 DATA .8,13,.3,15,.1,12
390 DATA .1,13,.1,20,.1,8,.1,20,.1,10,.1,20,
    .1,12,.1,20,.1,13,.1,20,.1,15,.1,20
400 DATA .1,17,.1,20,.1,25,.1,24,.1,22,.1,20,
    .1,18,.1,17,.1,15,.1,20,.1,18,.1,15
410 DATA .1,13,.1,20,.1,8,.1,20,.1,10,.1,20,
    .1,12,.1,20,.1,13,.1,20,.1,15,.1,20
420 DATA .1,17,.1,20,.1,25,.1,24,.1,22,.1,20,
    .1,18,.1,17,.1,15,.1,20,.1,18,.1,15
430 DATA .1,17,.1,18,.1,17,.1,16,.1,17,.1,12,
    .1,17,.1,16,.1,17,.1,12,.1,17,.1,16
440 DATA .6,17,.2,12,.2,17,.2,16
450 DATA .6,17,.2,12,.6,17
460 DATA .2,16,.6,17,.2,16
470 RESTORE 60
480 FOR i=1 TO 56
```

```
490 READ a,b
500 BEEP a,b
510 NEXT i
520 RESTORE 50
530 FOR i=1 TO 34
540 READ a,b
550 BEEP a,b
560 NEXT i
570 RESTORE 620
580 FOR i=1 TO 77
590 READ a,b
600 BEEP a,b
610 NEXT i
620 DATA 1.2,10,1.2,14
630 DATA .8,15,.2,17,.2,18
640 DATA .8,18,.4,18
650 DATA 1.2,17
660 DATA .8,15,.2,13,.2,12
670 DATA .8,10,.4,10
680 DATA .4,10,.4,13,.4,12
690 DATA 1.2,10,1.2,14
700 DATA .8,15,.2,17,.2,18
710 DATA .8,18,.4,18
720 DATA 1.2,18
730 DATA .8,16,.2,15,.2,13
740 DATA .8,11,.4,10
750 DATA .8,9,.4,9
760 DATA 1.2,10,1.2,12
770 DATA .133,-2,.133,1,.133,5,.133,10,.133,
   13,.133,17,.133,15,.133,13,.133,12
780 DATA .133,10,.133,13,.133,17,.133,22,.133,
   25,.133,29,.133,27,.133,25,.133,24
790 DATA .133,22,.133,25,.133,29,.133,34,.133,
   37,.133,41,.133,39,.133,37,.133,36
800 DATA .133,35,.133,34,.133,33,.133,32,.133,
   31,.133,30,.133,29,.133,28,.133,27
810 DATA .133,26,.133,25,.133,24,.133,23,.133,
   22,.133,21,.133,20,.133,19,.133,18
820 RESTORE 60
830 FOR i=1 TO 56
840 READ a,b
850 BEEP a,b
860 NEXT i
870 RESTORE 50
880 FOR i=1 TO 35
890 READ a,b
900 BEEP a,b
910 NEXT i
```

Programa 4.1.

Si nos fijamos podemos observar que el esquema utilizado para la realización del programa es similar al seguido en los programas del capítulo anterior. En líneas DATA se han almacenado parejas de datos que indican la duración y frecuencia de cada nota. El primero (líneas 50 a 200) corresponde al tema principal, y más conocido de la obra. El segundo grupo (líneas 310 a 460) corresponde a lo que sería el segundo tiempo, tras el cual vuelve a interpretarse el tema principal. Finalmente, el tercer grupo (líneas 620 a 810) está asociado al tercer tiempo, concluyendo definitivamente la obra con la interpretación por tercera vez del tema principal.

La interpretación se realiza con una serie de bucles FOR-NEXT que leen cada pareja de datos de las líneas DATA y ejecutan la nota correspondiente. El bucle de las líneas 270-800 es el encargado de interpretar el segundo tiempo, mientras que el de las líneas 580-610 ejecuta el tercer tiempo. El resto de los bucles interpretan el tema principal todas las veces que es necesario, mandando el puntero de datos mediante la instrucción RESTORE a la línea DATA adecuada.

Bueno, la idea no es nada complicada, sólo es necesario un poco de paciencia para ir traduciendo cada nota de la partitura a una pareja de números.



CONCIERTO A DOS VOCES PARA AMSTRAD

Aunque hasta ahora sólo hemos utilizado el canal A del AMSTRAD para nuestras experiencias musicales, sabemos que podemos utilizar simultáneamente hasta tres canales de sonido.

Por tanto, el primer parámetro de la instrucción SOUND no tiene como único objeto la mera selección de uno de los tres canales, sino que también le indica el ordenador si deben sonar varios canales al unísono o separadamente, si debe retener el sonido de un determinado canal o si debe borrar una cola de sonidos de alguno de los canales. La figura 4.1 muestra una tabla con los valores que toma el primer parámetro de SOUND dependiendo del efecto deseado.

Sin embargo, en el capítulo 2 dijimos que este primer parámetro podía tomar cualquier valor comprendido entre 1 y 255, lo que significa que podemos sumar varios de los valores representados en la tabla, consiguiendo de este modo combinar varios efectos. Por ejemplo, si queremos enviar un sonido al canal A sincronizándolo con el B el parámetro será 17 ya que:

1 (seleccionar canal A) + 16 (sincronizar con canal B) = 17

Por otra parte, si utilizamos el parámetro 64 produciremos la retención

Valor	Canal(es)	Efecto
1	A	Seleccionar canal A
2	B	Seleccionar canal B
4	C	Seleccionar canal C
8	→ A	Sincronizar con canal A
16	→ B	Sincronizar con canal B
32	→ C	Sincronizar con canal C
64		Retener sonido
128		Borrar cola de sonidos

Fig. 4.1. Valores para la selección de canales en AMSTRAD.

de una cola de sonido. Para anular esta retención disponemos de la instrucción:

RELEASE C

donde *C* es un parámetro comprendido entre 1 y 7 que indica el canal o

canales que deben liberarse. Por ejemplo la orden **RELEASE** 5 liberaría las colas de sonido retenidas en los canales A(1) y C(4).

Siguiendo con los parámetros de **SOUND**, habíamos dejado sin estudiar los tres últimos. Vamos a ver en este capítulo los correspondientes a la envolvente de volumen y la envolvente de tono.



La envolvente de volumen

Para definir una envolvente de volumen utilizaremos la instrucción ENV que tiene el siguiente formato:

ENV N, E1, A1, D1, E2, A2, D2, E3, A3, D3, E4, A4, D4, E5, A5, D5

donde *N* es el número de la envolvente definida, comprendido entre 1 y 15, y que será el número de referencia que citaremos más adelante como quinto parámetro de la instrucción SOUND correspondiente. Los quince parámetros restantes se pueden agrupar en realidad en tres que son los que constituyen una sección de la envolvente, lo que significa que cada envolvente puede disponer de hasta cinco secciones. El parámetro *En* indica el número de escalones de la sección correspondiente, es decir, en cuántas etapas de volumen queremos que evolucione la nota antes de concluir. Está comprendido entre 0 y 127.

An define la altura de cada escalón, de modo que el nivel de volumen de un escalón puede variar respecto al anterior en un número de unidades comprendido entre 0 y 15 (positivo o negativo). Sin embargo, los parámetros admisibles van de −128 a +127.

Finalmente, *Dn* especifica la duración de cada escalón en centésimas de segundo. El margen de valores va desde 0 a 255 lo que significa que la duración máxima de cada escalón es de 2.56 segundos.

Por consiguiente, es lógico que la duración total de la envolvente de volumen no sobrepase el valor especificado en el parámetro *duración* de la instrucción SOUND correspondiente, ya que en este caso el sonido terminaría antes de completar todos los escalones de la envolvente. Si por el contrario, la duración de la envolvente es más corta que el parámetro *duración,* la nota continuará sonando, aunque haya finalizado la envolvente, al nivel del último escalón.

El programa 4.2 es un ejemplo práctico de sonido con envolvente de volumen definida con cinco secciones.

El sonido producido está controlado por la instrucción ENV. En la instrucción SOUND hemos definido el canal A(1) y el tono (478). Sin embargo, la duración especificada es 0, lo que indica a la máquina que la duración de la nota es la definida en ENV. También se ha dado valor 0 al vo-

```
10 REM ******************
20 REM * EJEMPLO DE ENV *
30 REM *     AMSTRAD    *
40 REM ******************
50 CLS
60 LOCATE 13,13:PRINT "EJEMPLO DE ENV"
70 ENV 1,10,4,3,5,-3,20,1,0,20,5,3,10,10,-3,30
80 SOUND 1,478,0,0,1,0,0
```

Programa 4.2.

lumen ya que está especificado con más detalle en ENV. El quinto parámetro de SOUND es 1 que indica la envolvente que debe utilizar.

En la figura 4.2 podemos analizar la forma de la envolvente de volumen definida en el programa 4.2.

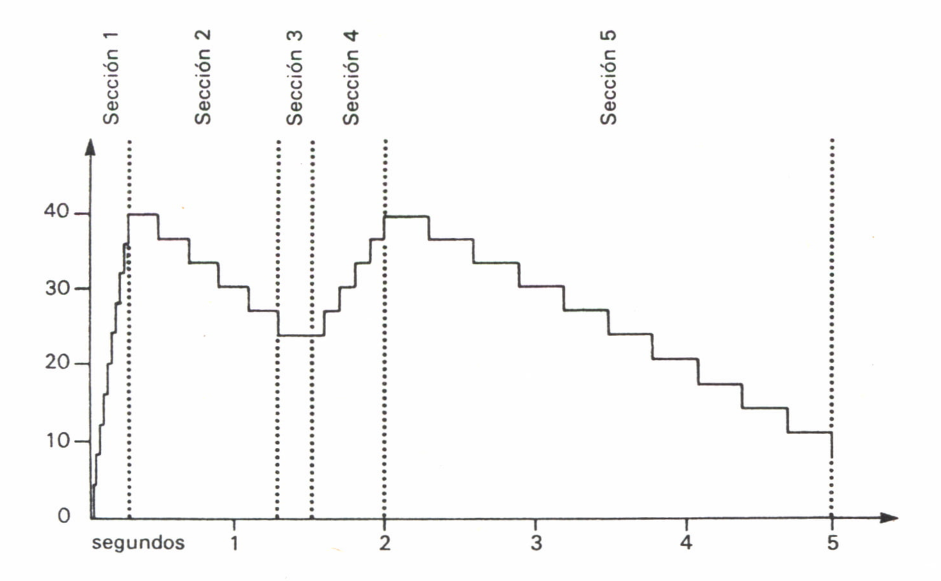


Fig. 4.2. Envolvente de volumen definida en el programa 4.2.

La envolvente de la figura 4.2 dura 5 segundos. En efecto, si multiplicamos el número de escalones por la duración de cada uno de ellos y sumamos los resultados para las cinco secciones obtenemos:

$$(3*10) + (20*5) + (20*1) + (10*5) + (30*10) = 500 \text{ centésimas} = 5 \text{ segundos}$$

Si queremos cancelar el efecto de una envolvente en la instrucción SOUND, debemos redefinirla sin especificar las secciones. Por ejemplo ENV1 desactiva la envolvente 1.

El programa 4.3 nos permite experimentar con distintas envolventes de volumen, incluyendo sólo una sección.

```
10 REM **************************
20 REM * ENVOLVENTES DE VOLUMEN *
30 REM *        AMSTRAD         *
40 REM **************************
50 CLS
60 INPUT "NUMERO DE SALTOS ";NS
70 INPUT "AMPLITUD DEL SALTO ";AS
80 INPUT "DURACION DEL SALTO ";DS
90 CLS
100 PRINT "NUMERO","AMPLITUD","DURACION"
110 PRINT NS,AS,DS
120 ENV 1,NS,AS,DS
130 SOUND 1,240,0,15,1,0,0
```

Programa 4.3.



La envolvente de tono

Para definir una envolvente de tono utilizamos la instrucción ENT, con el siguiente formato:

ENT N, E1, A1, D1, E2, A2, D2, E3, A3, D3, E4, A4, D4, E5, A5, D5

donde *N* es el número de la envolvente, que puede tomar cualquier valor entre 1 y 15, lo que significa que, al igual que con las envolventes de volumen, podemos llegar a definir hasta 15 envolventes diferentes. El parámetro N es el mismo que especificaremos en la instrucción SOUND correspondiente en sexta posición.

El resto de los parámetros de ENT siguen el mismo esquema que en ENV, es decir, se pueden definir entre una y cinco secciones para cada envolvente, cada una de las cuales consta de tres parámetros: *En,* que es el número de escalones, *An,* que es la altura de cada escalón, y *Dn,* que es la duración de cada escalón. Los márgenes de los valores son los mismos que para ENV excepto para En que va de 0 a 239.

Al igual que ocurre con ENV es necesario definir al menos una sección y siempre cada sección debe contar con todos los parámetros.

El programa 4.4 es un ejemplo del efecto de una envolvente de tono definida con cuatro secciones.

```
10 REM ******************
20 REM * EJEMPLO DE ENT *
30 REM *     AMSTRAD    *
40 REM ******************
50 CLS
60 LOCATE 13,13:PRINT "EJEMPLO DE ENT"
70 ENT 1,65,5,1,10,-2,10,10,2,5,30,-5,1
80 SOUND 1,478,50,15,0,1,0
```

Programa 4.4.

La figura 4.3 describe de modo esquemático el funcionamiento de ENT y SOUND en el programa 4.4.

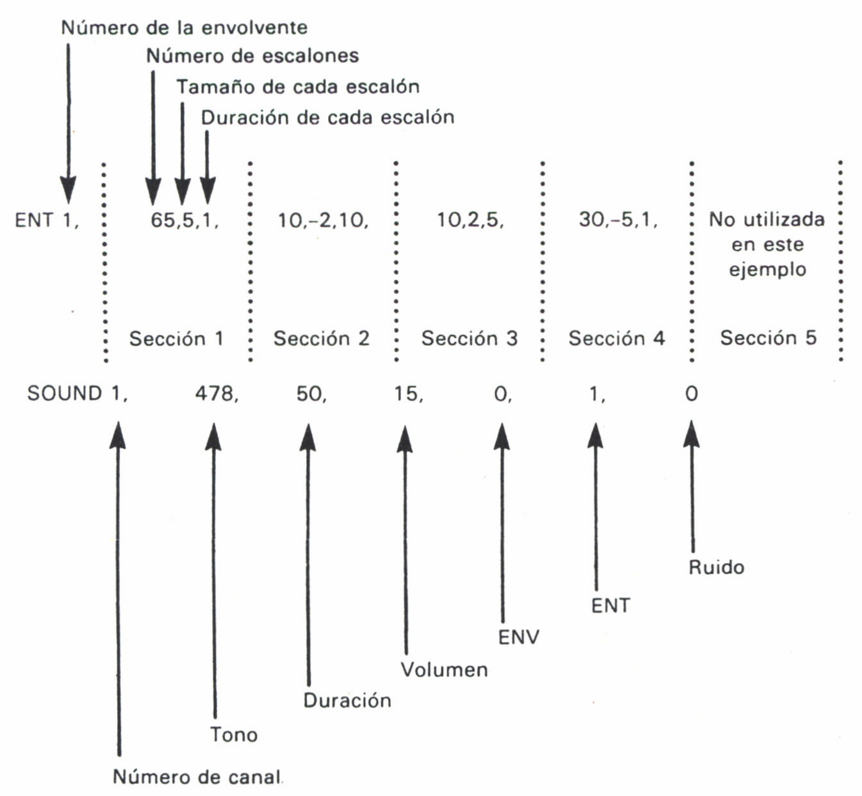


Fig. 4.3. Relación entre las instrucciones ENT y SOUND y secciones de ENT.

Lógicamente, si ENV representaba la variación del volúmen de una nota con el tiempo, ENT se encarga de definir la variación del tono de una nota a lo largo del tiempo. Esto está relacionado con las fases de ataque, caída, sostenimiento y relajación que vimos en el capítulo 2 cuando hablamos de los sonidos en el COMMODORE.

Finalmente, el programa 4.5 nos permite experimentar con distintas envolventes de tono con una sola sección.

```
10 REM ************************
20 REM * ENVOLVENTES DE TONO *
30 REM *       AMSTRAD        *
40 REM ************************
50 CLS
60 INPUT "NUMERO DE SALTOS ";NS
70 INPUT "AMPLITUD DEL SALTO ";AS
80 INPUT "DURACION DEL SALTO ";DS
90 CLS
100 PRINT "NUMERO","AMPLITUD","DURACION"
110 PRINT NS,AS,DS
120 ENT 1,NS,AS,DS
130 SOUND 1,240,100,15,0,1,0
```

Programa 4.5.



El concierto

Por fin, vamos a basarnos en todos los conceptos vistos anteriormente para desarrollar el programa 3.6 que interpreta un bonito concierto a dos voces utilizando la sincronización de los canales A y B. Las líneas DATA 220 y 230 almacenan la información referente a los tonos de las notas que componen la melodía en notación hexadecimal, mientras que las líneas DTA 240 a 255 almacenan toda la información sobre nota, octava, duración y sincronización para agudos y graves.

Para la interpretación de la melodía se definen dos envolventes de volumen, con tres secciones cada una (líneas 140 y 150), y una envolvente de tono, también con tres secciones (línea 160). La subrutina 260 es la encargada de enviar el sonido correspondiente al canal A mientras que la subrutina 440 envía el sonido al canal B.

```
10 REM *************************
20 REM * CONCIERTO A DOS VOCES *
30 REM *        AMSTRAD        *
40 REM *************************
50 DIM ESCALA(12)
60 FOR I=1 TO 12
70 READ ESCALA(I)
80 NEXT
90 CANAL1=1:READ C1$,C2$,c3$:CANAL1$=C1$+C2$+c3$
100 CANAL2=2:READ C4$,C5$:CANAL2$=C4$+C5$
110 CLS:LOCATE 10,13:PRINT "CONCIERTO A DOS VOCES"
120 VELOCIDAD=12
130 ESCALA$=" a-b b c+c d-e e f+f g+g"
140 ENV 1,2,5,2,8,-1,10,10,0,15
150 ENV 2,2,7,2,12,-1,10,10,0,15
160 ENT 1,1,1,1,2,-1,1,1,1,1
170 DEF FN M$(S$,S)=MID$(S$,S,1)
180 CANAL1=1:GOSUB 500
190 CANAL2=1:GOSUB 680
200 IF CANAL1+CANAL2>0 THEN GOTO 200
210 END
220 DATA &777,&70C,&6A7,&647,&5ED,&598
230 DATA &547,&4FC,&4B4,&470,&431,&3F4
240 DATA 4cr4f4f1f1g1A1-B2C2f4g2g1A1-B6A2Cr1f1g1f1g1a1-b1A1-
245 DATA b2C2g2A2g2f1g1a2g2f6e2c2e2c2g2e2c1-B1A2g2f4e4d8c4
247 DATA f3f1c2d4-b2fr2-B2A2g2f6e2gr4C4-B1a1f1-B1g2C2-B4A4g8f.
250 DATA r4f4f8f4e4c4fr8f4e2f2e4d2e2d8c8c6e2f4g4g8e4f3f1c4
255 DATA dr8g4cr4e4c6f2d4c4c8fr8-e4dr8g8c4e4c6f2d4c4c8f.
500 REM * ENVIAR SONIDO AL CANAL A *
510 P1$=FN M$(CANAL1$,CANAL1)
520 IF P1$<>"r" THEN R1=0:GOTO 540
530 R1=16:CANAL1=CANAL1+1:P1$=FN M$(CANAL1$,CANAL1)
540 IF P1$="." THEN CANAL1=0:RETURN ELSE L1=VAL(P1$)
550 CANAL1=CANAL1+1
560 N1$=FN M$(CANAL1$,CANAL1)
570 CANAL1=CANAL1+1
580 IF N1$="+" OR N1$="-" THEN GOTO 650
590 N1$=" "+N1$
600 ND1=(1+INSTR(ESCALA$,LOWER$(N1$)))/2
610 IF ASC(RIGHT$(N1$,1))>96 THEN O1=8 ELSE O1=16
620 SOUND 1+R1,ESCALA(ND1)/O1,VELOCIDAD*L1,0,1,1
630 ON SQ(1) GOSUB 500
640 RETURN
650 N1$=N1$+FN M$(CANAL1$,CANAL1)
660 CANAL1=CANAL1+1
670 GOTO 600
680 REM * ENVIAR SONIDO AL CANAL B *
690 P2$=FN M$(CANAL2$,CANAL2)
700 IF P2$<>"r" THEN R2=0:GOTO 720
710 R2=8:CANAL2=CANAL2+1:P2$=FN M$(CANAL2$,CANAL2)
720 IF P2$="." THEN CANAL2=0:RETURN ELSE L2=VAL(P2$)
730 CANAL2=CANAL2+1
740 N2$=FN M$(CANAL2$,CANAL2)
750 CANAL2=CANAL2+1
```

```
760 IF N2$="+" OR N2$="-" THEN GOTO 830
770 N2$=" "+N2$
780 ND2=(1+INSTR(ESCALA$,LOWER$(N2$)))/2
790 IF ASC(RIGHT$(N2$,1))>96 THEN O2=4 ELSE O2=8
800 SOUND 2+R2,ESCALA(ND2)/O2,VELOCIDAD*L2,0,1,2
810 ON SQ(2) GOSUB 680
820 RETURN
830 N2$=N2$+FN M$(CANAL2$,CANAL2)
840 CANAL2=CANAL2+1
850 GOTO 780
```

Programa 4.6.

EL COMMODORE NAVIDEÑO

El **COMMODORE** también es capaz de interpretar melodías utilizando varias voces. El programa 4.7 reproduce el popular villancico «Noche de Paz» a dos voces, utilizando los dos primeros canales de sonido. El esquema seguido en este programa es análogo al de los programas para el **COMMODORE** del capítulo anterior. Toda la información sobre notas y duración se almacena en líneas **DATA**, mientras que los **POKE** oportunos en las direcciones de memoria adecuadas se encargan de interpretar la melodía.

```
10 REM ******************************
20 REM * NOCHE DE PAZ (VILLANCICO) *
30 REM *          COMMODORE         *
40 REM ******************************
50 PRINT CHR$(147)
60 PRINT SPC(214);"NOCHE DE PAZ"
70 PRINT SPC(215);"VILLANCICO"
80 SID=54272
90 FOR I=SID TO SID+24
100 POKE I,0
110 NEXT I
120 DIM A(10),B(10)
130 FOR I=0 TO 10
140 READ A(I),B(I)
150 NEXT I
160 DIM AA(7),BB(7)
170 FOR I=0 TO 7
180 READ AA(I),BB(I)
190 NEXT I
200 POKE SID+24,15
210 POKE SID+6,240
```

```
220 POKE SID+13,64
240 IF D>0 THEN GOTO 310
250 READ D
260 IF D<0 THEN END
270 READ N
280 POKE SID+4,17
290 POKE SID,B(N)
300 POKE SID+1,A(N)
310 IF DD>0 THEN GOTO 380
320 READ DD
330 IF DD<0 THEN END
340 READ N
350 POKE SID+11,17
360 POKE SID+7,BB(N)
370 POKE SID+8,AA(N)
380 FOR I=1 TO 135:NEXT I
390 IF D=1 THEN POKE SID+4.16
400 IF DD=1 THEN POKE SID+11,16
410 FOR I=1 TO 15:NEXT I
420 D=D-1:DD=DD-1
430 GOTO 240
440 DATA 21,154,22,227,25,177,28,214,32,94
450 DATA 34,75,38,126,43,52,45,198,19,63,17,37
460 DATA 14,24,8,97,10,143,12,143
470 DATA 11,48,6,71,7,233,9,104
480 DATA 3,2,2,1,2,2,1,3,2,2,2,3
490 DATA 6,0,2,1,2,2,2,3
500 DATA 3,2,2,1,2,2,1,3,2,2,2,3
510 DATA 6,0,2,1,2,2,2,3
520 DATA 4,6,2,5,2,6,2,6,2,7
530 DATA 6,4,2,5,2,6,2,7
540 DATA 4,5,2,1,2,2,2,5,2,3
550 DATA 6,2,2,1,2,2,2,3
560 DATA 4,3,2,1,2,4,2,3,2,0
570 DATA 3,5,2,1,2,4,1,4,2,3,2,0
580 DATA 3,2,2,1,2,2,1,3,2,2,2,3
590 DATA 6,0,2,1,2,2,2,3
600 DATA 4,3,2,1,2,4,2,3,2,0
610 DATA 3,5,2,1,2,4,1,4,2,3,2,0
620 DATA 3,2,2,1,2,2,1,3,2,2,2,3
630 DATA 6,0,2,1,2,2,2,3
640 DATA 4,6,2,5,2,6,2,6,2,7
650 DATA 3,8,2,5,2,61,6,2,5,2,7
660 DATA 6,5,2,1,2,2,2,3
670 DATA 6,7,2,1,2,2,2,3
680 DATA 3,5,2,1,2,2,1,2,2,0,2,3
690 DATA 3,2,2,1,2,2,1,1,2,9,2,3
700 DATA 12,10,12,3,-1
```

Programa 4.7.



SELECCION MUSICAL PARA IBM

Por último, incluimos aquí un espectacular programa que nos permite elegir la melodía que deseamos escuchar, sólo con pulsar una tecla. La presentación en pantalla resulta muy atrayente ya que aparece el teclado de un piano en que se va indicando la nota que suena en cada momento.

```
10 REM ****************************
20 REM * SELECCCION DE MELODIAS *
30 REM *            IBM           *
40 REM ****************************
50 SCREEN 0,1:WIDTH 40:COLOR 15,1,1:
CLS:DEFINT A-Z
60 LOCATE 15,7:PRINT " ------- selec
ciones -------"
70 LOCATE 16,7:PRINT " A-MARCH  E-HUMOR
I-SAKURA"
80 LOCATE 17,7:PRINT " B-STARS  F-BUG
J-BLUE  "
90 LOCATE 18,7:PRINT " C-FORTY  G-POP
K-SCALES"
100 LOCATE 19,7:PRINT " D-HAT    H-DANDY
ESC-SALIDA"
110 COLOR 15,0
120 FOR I=0 TO 15:FOR J=0 TO 8
130 LOCATE 5+J,5+I*2:PRINT CHR$(219);
CHR$(221);
140 NEXT:NEXT
150 FOR I=0 TO 12:FOR J=0 TO 4
160 IF I=2 OR I=6 OR I=9 OR I=13 THEN 180
170 LOCATE 5+J,8+I*2:PRINT CHR$(32);
CHR$(222);
180 NEXT:NEXT
190 FOR J=0 TO 9
200 LOCATE 4+J,4:COLOR 4,0:PRINT CHR$(221);
:LOCATE 4+J,36:COLOR 15,0
210 PRINT CHR$(221);:COLOR 4,1:PRINT
CHR$(221);
220 NEXT
230 COLOR 4,1:LOCATE 4,4
240 FOR I=0 TO 32:PRINT CHR$(219);:NEXT
250 PRINT CHR$(221);:LOCATE 13,4
260 FOR I=0 TO 32:PRINT CHR$(219);:NEXT
270 PRINT CHR$(221);:COLOR 0,7:DIM M(88),
O(70)
280 FOR I=7 TO 88:M(I) =  36.8*(2^(1/12))^
(I-6):NEXT
290 FOR I=0 TO 6:M(I) = 32767:NEXT
```

```
300 O(0) = 0
310 O(39)=5:O(40)=7:O(41)=8:O(42)=9
320 O(43)=10:O(44)=11:O(45)=13:O(46)=14
330 O(47)=15:O(48)=16:O(49)=17:O(50)=18
340 O(51)=19:O(52)=21:O(53)=22:O(54)=23
350 O(55)=24:O(56)=25:O(57)=27:O(58)=28
360 O(59)=29:O(60)=30:O(61)=31:O(62)=32
370 O(63)=33:O(64)=35:O(65)=36:O(66)=37
380 O(67)=38:O(68)=39:O(69)=40:O(70)=42
390 GOTO 530
400 READ J,K
410 CMD$ = INKEY$:IF CMD$="" THEN 430
420 IF CMD$=CHR$(27) THEN RETURN
430 IF J = -1   THEN RETURN
440 Q = O(J)
450 IF J>64 OR J<39 THEN 480
460 IF SCREEN(5,Q)<>32 THEN COLOR 0,7:
LOCATE 11,Q:PRINT CHR$(14);:COLOR 15,0:
GOTO 480
470 COLOR 15,0:LOCATE 7,Q:PRINT CHR$(14)
;:COLOR 0,7
480 SOUND M(J),K:IF J=0 AND K=1 THEN 500
490 SOUND 32767,1
500 IF J>64 OR J<39 THEN 400
510 IF SCREEN(5,Q) = 32 THEN COLOR 15,0:
LOCATE 7,Q:PRINT CHR$(32);:GOTO 400
520 COLOR 15,0:LOCATE 11,Q:PRINT
CHR$(219);:GOTO 400
530 LOCATE 21,5:PRINT "
                              ";
540 LOCATE 21,5:PRINT "SELECCION DE
MELODIA==>";
550 IF INKEY$ <> "" THEN 550
560 CMD$=INKEY$:IF CMD$="" THEN 560
570 IF CMD$=CHR$(27) THEN GOTO 780
580 IF CMD$="A" OR CMD$="a" THEN S$=
"MARCH ":RESTORE 1420:GOTO 700
590 IF CMD$="B" OR CMD$="b" THEN S$=
"STARS ":RESTORE 1510:GOTO 700
600 IF CMD$="C" OR CMD$="c" THEN S$=
"FORTY ":RESTORE 1200:GOTO 700
610 IF CMD$="D" OR CMD$="d" THEN S$=
"HAT   ":RESTORE 1620:GOTO 700
620 IF CMD$="E" OR CMD$="e" THEN S$=
"HUMOR ":RESTORE 1040:GOTO 700
630 IF CMD$="F" OR CMD$="f" THEN S$=
"BUG   ":RESTORE 790:GOTO 700
640 IF CMD$="G" OR CMD$="g" THEN S$=
"POP   ":RESTORE 1120:GOTO 700
650 IF CMD$="H" OR CMD$="h" THEN S$=
```

```
"DANDY ":RESTORE 1320:GOTO 700
660 IF CMD$="I" OR CMD$="i" THEN S$=
"SAKURA":RESTORE 1780:GOTO 700
670 IF CMD$="J" OR CMD$="j" THEN S$=
"BLUE  ":RESTORE 870:GOTO 700
680 IF CMD$="K" OR CMD$="k" THEN S$=
"SCALES":RESTORE 1700:GOTO 700
690 GOTO 540
700 PRINT " ";CMD$;"-";S$
710 READ D
720 READ S$:LOCATE 23,1+(40.5-LEN(S$))/2
730 COLOR 15,4:PRINT S$;:COLOR 0,7
740 GOSUB 400
750 S$=STRING$(39," "):LOCATE 23,1:
COLOR 4,1:PRINT S$:COLOR 0,7
760 GOTO 530
770 END
780 SCREEN 0,1:COLOR 7,0,0:CLS:END
790 DATA -2,"La Cucaracha -Cancion
popular mexicana"
800 DATA 42,1,0,1,42,1,0,1,42,1,0,1,47,
1,0,5,51,1,0,3,42,1,0,1,42,1,0,1
810 DATA 42,1,0,1,47,1,0,5,51,1,0,5,30,
1,0,1,30,1,0,1,35,1,0,3,47,1,0,1
820 DATA 47,1,0,1,46,1,0,1,46,1,0,1,44,
1,0,1,44,1,0,1,42,8,0,2,42,1,0,1
830 DATA 42,1,0,1,42,1,0,1,46,1,0,5,49,
1,0,3,42,1,0,1,42,1,0,1,42,1,0,1
840 DATA 46,1,0,5,49,1,0,5,37,1,0,1,37,
1,0,1,30,1,0,3,54,2,56,2,54,2,52,2
850 DATA 51,2,49,2,47,8
860 DATA -1,-1
870 DATA -2,"Vals Danubio Azul por J.S.
Strauss"
880 DATA 42,4,46,4,49,4,49,4,0,4,61,2,0,
2,61,2,0,6,58,2,0,2,58,2,0,6,42,4,42,4
890 DATA 46,4,49,4
900 DATA 49,4,0,4,61,2,0,2,61,2,0,6,59,2,
0,2,59,2,0,6,41,4,41,4,44,4,51,4,51,4
910 DATA 0,4,63,2,0,2,63,2,0,6,59,2,0,2
920 DATA 59,2,0,6,41,4,41,4,44,4,51,4,51,
4,0,4,63,2,0,2,63,2,0,6,58,2,0,2,58,2
930 DATA 0,6,42,4
940 DATA 42,4,46,4,49,4,54,4,0,4,66,2,0,2,
66,2,0,6,61,2,0,2,61,2,0,6,42,4
950 DATA 42,4,46,4,49,4,54,4,0,4,66,2,0,2
960 DATA 66,2,0,6,63,2,0,2,63,2,0,6,44,4,
44,4,47,4,51,2,0,2,51,14,0,2,48,4
970 DATA 49,4,58,16
980 DATA 54,4,46,4,46,8,44,4,51,8,49,4,42,
```

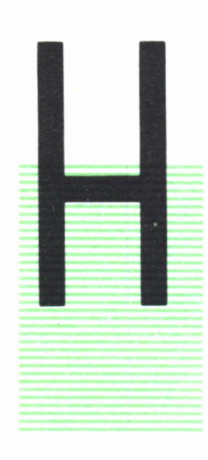
```
4,0,2,42,2,42,4,0,8,49,2,0,2,47,2
990 DATA 0,6,49,2,0,2
1000 DATA 47,2,0,6,49,4,58,16,56,4,49,2,0,
2,46,2,0,6,49,2,0,2,46,2,0,6,49,4
1010 DATA 56,16,54,4,49,2,0,2,47,2,0,6,49,
2,0,2,47,2,0,6,49,4,58,16
1020 DATA 56,4,49,4,54,4,56,4,58,4,61,8,59,
4,58,2,58,2,58,4,56,2,0,2,54,4,0,8
1030 DATA -1,-1
1040 DATA -2,"Humoresque por Dvorak"
1050 DATA 47,3,0,2,49,1,47,3,0,2,49,1,51,
3,0,2,54,1,56,3,0,2,54,1
1060 DATA 59,3,0,2,58,1,61,3,0,2,59,1,58,
3,0,2,61,1,59,3,0,2,56,1
1070 DATA 54,3,0,2,54,1,56,3,0,2,54,1,
59,3,0,2,56,1,54,3,0,2,51,1
1080 DATA 49,24,47,3,0,2,49,1,47,3,0,2,
49,1,51,3,0,2,54,1,56,3,0,2,54,1
1090 DATA 56,3,0,2,58,1,61,3,0,2,59,1,
58,3,0,2,61,1,59,3,0,2,56,1
1100 DATA 54,3,0,2,54,1,59,3,0,2,47,1,
49,6,54,6,47,18
1110 DATA -1,-1
1120 DATA -2,"Pop! Goes the Weasel -
Anonimo"
1130 DATA 47,2,0,2,47,2,49,2,0,2,49,2,
51,2,54,2,51,2,47,2,0,2,42,2
1140 DATA 47,2,0,2,47,2,49,2,0,2,49,2,
51,6,47,2,0,2,42,2,47,2,0,2,47,2,49,2
1150 DATA 0,2,49,2,51,2,54,2,51,2,47,2,
0,4,56,2,0,4,49,2,0,2,52,2,51,6,47,2
1160 DATA 0,4,59,2,0,2,59,2,56,2,0,2,59,
2,58,2,61,2,58,2,54,2,0,4,59,2,0,2
1170 DATA 59,2,56,2,0,2,59,2,58,6,54,2,
0,2,51,2,52,2,0,2,51,2,52,2,0,2,54,2
1180 DATA 56,2,0,2,58,2,59,2,0,4,56,2,0,
4,49,2,0,2,52,2,51,6,47,2
1190 DATA -1,-1
1200 DATA -2,"Sinfonia 40 por Mozart"
1210 DATA 55,2,54,2,54,4,55,2,54,2,54,4,
55,2,54,2,54,4,62,4,0,4
1220 DATA 62,2,61,2,59,4,59,2,57,2,55,4,
55,2,54,2
1230 DATA 52,4,52,4,0,4,54,2,52,2,52,4,
54,2,52,2,52,4,54,2,52,2
1240 DATA 52,4,61,4,0,4,61,2,59,2,58,4,
58,2,55,2,54,4,54,2,52,2
1250 DATA 50,4,50,4,0,4,62,2,61,2,61,4,
64,4,58,4,61,4
1260 DATA 59,4,54,4,0,4,62,2,61,2,61,4,
```

```
64,4,58,4,61,4
1270 DATA 59,4,62,4,61,2,59,2,57,2,55,2,
54,4,46,4,47,4,49,4
1280 DATA 50,4,52,2,50,2,49,4,47,4,54,4,
0,4,65,8
1290 DATA 66,2,0,6,65,8,66,2,0,6,65,8
1300 DATA 66,4,65,4,66,4,65,4,66,4
1310 DATA -1,-1
1320 DATA -2,"Yankee Doodle - Anonimo "
1330 DATA 50,3,50,3,52,3,54,3,50,3,54,3,
52,3,45,3,50,3,50,3,52,3,54,3,50,6
1340 DATA 49,3,0,3
1350 DATA 50,3,50,3,52,3,54,3,55,3,54,3,
52,3,50,3,49,3,45,3,47,3,49,3,50,6
1360 DATA 50,3,0,3
1370 DATA 47,5,49,1,47,3,45,3,47,3,49,3,
50,3,0,3,45,5,47,1,45,3,43,3,42,6
1380 DATA 45,3,0,3
1390 DATA 47,5,49,1,47,3,45,3,47,3,49,3,
50,3,47,3,45,3,50,3,49,3,52,3,50,6
1400 DATA 50,6
1410 DATA -1,-1
1420 DATA -2,"Marcha Funebre de una
Marioneta-Gounod"
1430 DATA 37,1,0,2,30,1,0,5,42,3,42,3,41,
3,39,3,41,3,0,3,42,3,44,3,0,3,37,1,0,2
1440 DATA 30,1,0,5,42,3,42,3,41,3,39,3,41,
3,0,3,42,3,44,3,0,3,37,3,42,3,0,3,45,3
1450 DATA 49,6,47,3,45,3,0,3,49,3,52,6,50,
3,49,3,0,3,53,3,56,6,54,3,53,3,50,3
1460 DATA 49,3,47,3,45,3,44,3,30,1,0,5,42,
3,42,3,41,3,39,3,41,3,0,3,42,3,44,3
1470 DATA 0,3,37,1,0,2,30,1,0,5,42,3,42,3,
41,3,39,3,41,3,0,3,42,3,44,3,0,3
1480 DATA 37,3,45,3,0,3,49,3,52,6,50,3,49,
3,47,3,45,3,43,3,47,3,50,3,42,3
1490 DATA 41,3,42,3,44,3,0,3,45,1,0,2,44,
9,42,1
1500 DATA -1,-1
1510 DATA -2,"Stars and Stripes Forever -
Sousa "
1520 DATA 54,6,54,6,52,3,51,3,51,6,50,3,
51,3,51,16,0,2,50,3
1530 DATA 51,3,51,6,50,3,51,3
1540 DATA 54,6,51,3,54,3,52,12,49,6,0,3,
49,3,49,6,48,3,49,3
1550 DATA 49,6,48,3,49,3
1560 DATA 52,16,0,2,51,3,49,3,51,3,54,9,
56,9,56,3,49,16,0,2,54,6
1570 DATA 54,6,52,3,51,3,51,6,50,3,51,3,
```

```
51,16,0,2,50,3,51,3,51,6,50,3 ,51,3
1580 DATA 52,3,51,3,49,5,46,1,49,12,47,6,
0,3,47,3,47,6,46,3,47,3,50,6,49,3,47,3
1590 DATA 59,15,0,3,47,3,49,3,51,3,54,1,
0,2,47,3,49,3,51,3,54,1,0,2,42,3,44,5
1600 DATA 51,1,49,12,47,1
1610 DATA -1,-1
1620 DATA -2,"Baile del Sombrero Mexicano
-Popular"
1630 DATA 52,2,57,2,0,2,52,2,57,2,0,2,52,
2,57,6,0,4,52,2,57,2,59,2,57,2,56,4
1640 DATA 57,2,59,2,0,8,52,2,56,2,0,2,52,
2,56,2,0,2,52,2,56,6,0,4,52,2
1650 DATA 56,2,57,2,56,2,54,4,56,2,57,2,
0,6,64,2,63,2,64,2,61,2,60,2,61,2
1660 DATA 57,2,56,2,57,2,52,2,0,4,49,2,
50,2,52,2,54,2,56,2,57,2,59,2,61,2
1670 DATA 62,2,59,2,0,4,62,2,61,2,62,2,
59,2,58,2,59,2,56,2,55,2,56,2,52,2
1680 DATA 0,4,64,2,63,2,64,2,66,2,64,2,
62,2,61,2,59,2,57,2
1690 DATA -1,-1
1700 DATA -2,"Escalas"
1710 DATA 38,1,39,1,40,1,41,1,42,1,43,
1,44,1,45,1,46,1,47,1
1720 DATA 48,1,49,1,50,1,51,1,52,1,53,
1,54,1,55,1,56,1
1730 DATA 57,1,58,1,59,1,60,1,61,1,62,
1,63,1,64,1,65,8,0,4
1740 DATA 65,8,64,1,63,1,62,1,61,1,60,
1,59,1,58,1,57,1
1750 DATA 56,1,55,1,54,1,53,1,52,1,51,
1,50,1,49,1,48,1
1760 DATA 47,1,46,1,45,1,44,1,43,1,42,
1,41,1,40,1,39,1,38,8
1770 DATA -1,-1
1780 DATA -2,"Sakura - Melodia Popular
Japonesa"
1790 DATA 49,8,49,8,51,12,0,4,49,8,49,8,
51,12,0,4,49,8,51,8,52,8,51,8
1800 DATA 49,8,51,4,49,4,45,16,44,8,40,8,
44,8,45,8
1810 DATA 44,8,44,4,40,4,39,16,49,8,49,8,
51,12,0,4,49,8,49,8,51,12,0,4
1820 DATA 40,8,44,8,45,8,49,8,51,4,49,4,
45,8,44,16
1830 DATA -1,-1
```

Programa 4.8.

EL TECLADO CONVERTIDO EN PIANO 5



ASTA ahora el ordenador ha sido el único intérprete de todas las melodías y, desde luego, no lo ha hecho nada mal. Sin embargo, también sería interesante que nosotros pudiéramos interpretar las melodías que quisiéramos pero para ello necesitaríamos algún instrumento musical como, por ejemplo, un piano. En este capítulo vamos a ver cómo podemos transformar el teclado de nuestro ordenador en un piano que nos permita demostrar a los demás nuestras habilidades musicales.

EL PIANO-SPECTRUM

Podemos utilizar el teclado del SPECTRUM para que actúe del mismo modo que un teclado de piano, introduciendo notas en unas cuantas teclas seleccionadas. En el momento en que se pulse esa tecla determinada, sonará la nota correspondiente en el SPECTRUM.

El primer paso es elegir las teclas que se van a usar para producir las notas musicales. Lógicamente debemos intentar obtener un teclado similar al del piano. Por tanto, vamos a elegir la fila intermedia de letras, de la A a la L para las notas naturales (teclas blancas del piano), comenzando por la A que reproduce el DO medio. Algunas teclas de la fila superior, de la Q a la P reproducen las notas sostenidas (teclas negras del piano).

Para detectar qué tecla se ha tocado, utilizamos el comando INKEY$, que se asigna a la variable A$ en la línea 150. Si pulsamos una tecla distinta de las seleccionadas no se oirá nada ya que el control del programa volverá a la línea 150. Si por el contrario, pulsamos una de las teclas seleccionadas oiremos la tecla correspondiente. Para asignar el tono adecuado a cada tecla dimensionamos una matriz de 25 elementos (línea 70), cada uno de los cuales almacena una frecuencia. Los códigos de las teclas uti-

lizadas van desde el 65 (A) hasta el 89 (Y). En la instrucción BEEP, el código de la tecla pulsada se obtiene utilizando el comando CODE y restando 64 para obtener un número entre 1 y 25. De este modo se selecciona el elemento de la matriz correspondiente y por tanto la frecuencia asignada.

Los códigos entre 65 y 89 que no corresponden a ninguna de las teclas seleccionadas tienen asignada una frecuencia de 64 (almacenadas en líneas DATA) que produce un sonido demasiado agudo para que sea perceptible por el oído humano.

El programa 5.1 recoge todo lo explicado anteriormente para transformar el SPECTRUM en un piano. Además dibuja en pantalla un teclado de piano con las teclas correspondientes del SPECTRUM para que nos sea más fácil la interpretación.

```
 10 REM ************
 20 REM *   PIANO   *
 30 REM * SPECTRUM *
 40 REM ************
 50 CLS
 60 GO SUB 200
 70 DIM F(25)
 80 FOR I=1 TO 25
 90 READ F(I)
100 NEXT I
110 DATA 0,64,64,4,3,5
120 DATA 7,9,64,11,12,14
130 DATA 64,64,13,64,64,64
140 DATA 2,6,10,64,1,64,8
150 LET A$=INKEY$: IF A$="" THEN  GO TO 150
160 IF CODE A$<65 OR CODE A$>89 THEN  GO TO 150
170 BEEP 0.3,F(CODE A$-64)
180 GO TO 150
200 REM * DIBUJO DEL TECLADO *
210 LET B$=" "+CHR$ 133+CHR$ 138
220 FOR I=1 TO 4
230 PRINT AT 3+I,3;B$;
240 FOR J=1 TO 7
250 IF J=2 OR J=6 THEN  PRINT "   ";: NEXT J
260 PRINT B$;
270 NEXT J
280 NEXT I
290 FOR I=0 TO 8
300 PLOT 16+I*24,80
310 DRAW 24,0
320 DRAW 0,64
330 DRAW -24,0
```

```
340 DRAW 0,-64
350 NEXT I
360 PRINT AT 2,5;"W  E     T  Y  U     O"
370 PRINT AT 13,3;"A  S  D  F  G  H  J  K  L"
380 RETURN
```

Programa 5.1.

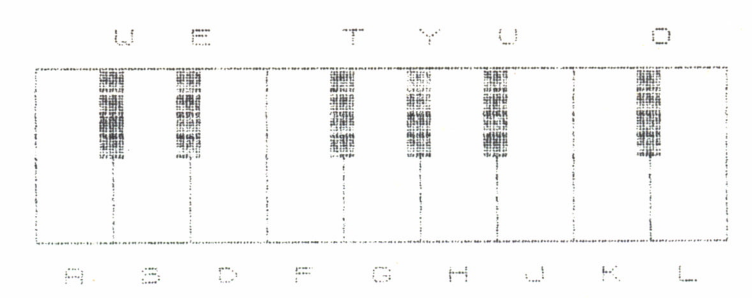


Fig. 5.1. Pantalla empleada en el programa de utilización del SPECTRUM como piano.

Finalmente hay que añadir que siempre podemos ampliar el teclado a más escalas aumentando la matriz de frecuencias y las teclas seleccionadas.



EL PIANO-AMSTRAD E IBM

Vamos a transformar nuestro **AMSTRAD** en un piano que abarque tres octavas completas. Para ello vamos a utilizar casi todas las teclas. Las filas segunda y cuarta interpretan las notas naturales de las tres octavas, mientras que algunas de las teclas de las filas primera y tercera están asignadas a las notas sostenidas (teclas negras del piano). Podemos ver la relación entre el piano y el teclado del **AMSTRAD** en la figura 5.2.

En el programa 5.2 formamos una cadena con todas las teclas seleccionadas y la almacenamos en la variable **T$** (línea 70). A continuación un bucle **WHILE-WEND** nos permite tocar tantas notas como deseemos. En la línea 90 establecemos la función **INKEY$** que nos permite pulsar una tecla. Las condiciones de las líneas 100 a 117 se encargan de comprobar que esa tecla no está entre las no seleccionadas. En la línea 120 utilizamos la función **INSTR** para asignar a la variable **N** el valor de la posición que

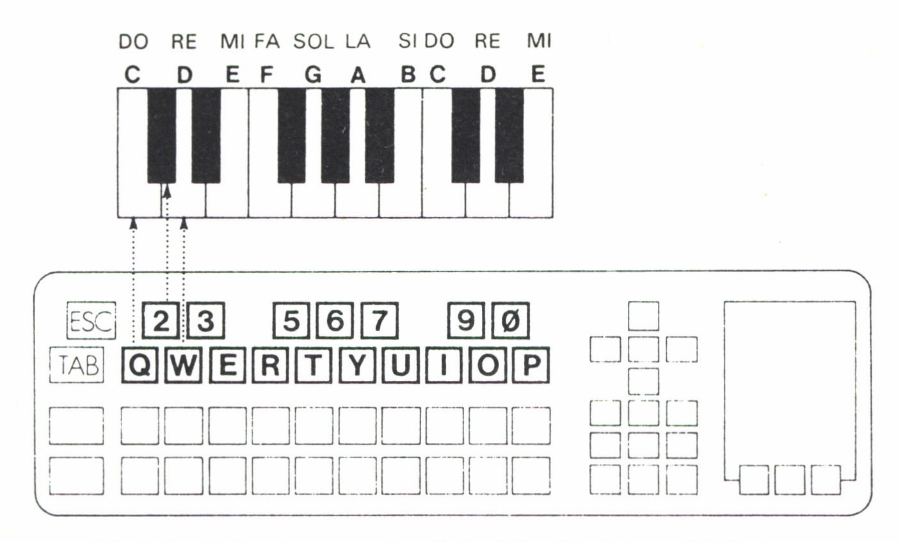


Fig. 5.2. Relación entre el teclado del AMSTRAD y el del piano.

ocupa la tecla pulsada en la cadena T$. Dicho valor N nos permite calcular la frecuencia de la nota (línea 130) y el tono (línea 140). Por último, la instrucción SOUND de la línea 150 se encarga de interpretar la nota.

Por otra parte, y al igual que en el programa anterior, la subrutina 180 dibuja en pantalla el teclado del piano con las tres octavas y las teclas del AMSTRAD asociadas.

```
10 REM ********************
20 REM * TECLADO DE PIANO *
30 REM *      AMSTRAD      *
40 REM ********************
50 MODE 1:CLS
60 GOSUB 180
70 T$="Q2W3ER5T6Y7UI9OOP@^[AZSXCFVGBNJMK,L./"
80 WHILE W=0
90 N$=INKEY$:IF N$="" THEN GOTO 90
100 IF N$="1" OR N$="4" OR N$="8" THEN GOTO 90
105 IF N$="-" OR N$="D" OR N$="H" THEN GOTO 90
107 IF N$="¦" OR N$=";" OR N$="]" OR N$="\" THEN GOTO 90
110 A=ASC(N$)
115 IF A=9 OR A=16 OR A=127 THEN GOTO 90
117 IF A=13 OR A=32 OR A=224 THEN GOTO 90
120 N=INSTR(T$,N$)
130 F=440*(2^(0+(N-10)/12))
140 T=ROUND(125000/F)
```

```
150 SOUND 1,T,30,15
160 WEND
170 END
180 REM * DIBUJO DEL TECLADO *
190 FOR I=10 TO 30 STEP 2
200 FOR J=4 TO 6
210 IF I=14 OR I=22 OR I=28 THEN GOTO 230
220 LOCATE I,J:PRINT CHR$(143)
230 NEXT J,I
240 MOVE 120,256
250 DRAWR 384,0
260 MOVE 120,350
270 DRAWR 384,0
280 FOR X=120 TO 504 STEP 32
290 MOVE X,256
300 DRAWR 0,94
310 NEXT
320 FOR I=10 TO 26 STEP 2
330 FOR J=14 TO 16
340 IF I=14 OR I=20 THEN GOTO 360
350 LOCATE I,J:PRINT CHR$(143)
360 NEXT J,I
370 MOVE 152,96
380 DRAWR 320,0
390 MOVE 152,190
400 DRAWR 320,0
410 FOR X=152 TO 472 STEP 32
420 MOVE X,96
430 DRAWR 0,94
440 NEXT
450 LOCATE 10,3:PRINT "2 3   5 6 7   9 0   ^"
460 LOCATE 9,10:PRINT "Q W E R T Y U I O P @ ["
470 LOCATE 10,13:PRINT "A S   F G   J K L"
480 LOCATE 11,20:PRINT "Z X C V B N M , . /"
490 RETURN
```

Programa 5.2.

En cuanto al piano IBM el programa sería análogo al anterior ya que dispone de las instrucciones WHILE-WEND, INKEY$, INSTR y SOUND (ésta con sólo dos parámetros). Puede variar un poco la subrutina de dibujo del teclado ya que en IBM el origen de coordenadas para alta resolución está en el ángulo superior izquierdo mientras que AMSTRAD está en el ángulo inferior izquierdo.



EL PIANO COMMODORE

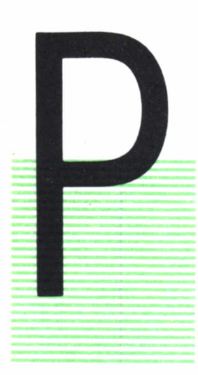
Para transformar el **COMMODORE** en un piano hemos seguido el mismo esquema que para el **SPECTRUM**, pero en este caso necesitamos dos matrices, una para altas frecuencias y otra para bajas. En las líneas **DATA** almacenamos en este caso parejas de números que representan la alta y baja frecuencia de cada nota. La función **INKEY$** es sustituida por **GET A$** que tiene la misma misión. Los **POKE** son los mismos que hemos utilizado hasta el momento y no necesitan ningún comentario.

```
10 REM ***************
20 REM *    PIANO    *
30 REM *  COMMODORE  *
40 REM ***************
50 PRINT CHR$(147);SPC(218);"PIANO"
60 DIM A(25),B(25)
70 FOR I=1 TO 25
80 READ A(I),B(I)
90 NEXT I
100 DATA 34,75,0,0,0,0,43,52,40,200
110 DATA 45,198,51,97,57,172,0,0,64,188
120 DATA 68,149,76,252,0,0,0,0,72,169
130 DATA 0,0,0,0,0,0,38,126,48,127
140 DATA 61,126,0,0,36,85,0,0,54,111
150 FOR I=54272 TO 54296
160 POKE I,0
170 NEXT I
180 POKE 54296,15
190 POKE 54277,52:POKE 54278,124
200 GET A$:IF A$="" THEN GOTO 200
210 IF ASC(A$)<65 OR ASC(A$)>89 THEN GOTO 200
220 POKE 54276,17
230 POKE 54272,B(ASC(A$)-64)
240 POKE 54273,A(ASC(A$)-64)
250 FOR I=1 TO 300:NEXT I
260 POKE 54276,16
270 GOTO 200
```

Programa 5.3.

Conviene indicar que las teclas seleccionadas para el piano son las mismas que en el **SPECTRUM**, es decir, de la **A** a la **L** para las notas naturales y de la **Q** a la **P** para las notas sostenidas.

SONIDOS ESPECIALES 6



ARA finalizar el tema de los sonidos con ordenador no podía faltar un capítulo dedicado a los sonidos especiales. Efectivamente, el ordenador no sólo es capaz de producir música sino también toda una variada serie de interesantes efectos sonoros que puede ir desde una explosión hasta un helicóptero, sin olvidar al «ordenador-compositor» capaz de generar música aleatoria.

EFECTOS SONOROS

Vamos a comenzar por el estudio de unos cuantos efectos sonoros en distintos ordenadores. Estos efectos suelen ser bastante interesantes de cara al diseño de videojuegos.

Comencemos por la generación de un trino:

```
 10 REM ************
 20 REM *  TRINOS  *
 30 REM * SPECTRUM *
 40 REM ************
 50 CLS
 60 PRINT AT 11,13;"TRINOS"
 70 FOR I=1 TO 4
 80 FOR J=1 TO 50
 90 BEEP 0.02,40
100 BEEP 0.02,40+I
110 NEXT J
120 PAUSE 25
130 NEXT I
```

Programa 6.1.

El programa 6.1 produce un tipo de sonido gorgojeante mediante la sucesión de un trino de dos sonidos. Podemos cambiar los valores de la frecuencia del segundo sonido y comprobar los efectos que produce.

El programa 6.2 produce el efecto de caída de una bomba que finalmente explota.

```
10 REM ******************
20 REM * CAIDA DE BOMBA *
30 REM *     AMSTRAD    *
40 REM ******************
50 CLS
60 LOCATE 19,13:PRINT "CAIDA"
70 FOR I=50 TO 150
80 SOUND 1,I,3,15,0,0,0
90 NEXT
100 LOCATE 16,13:PRINT "EXPLOSION'
110 FOR I=1 TO 100
120 SOUND 1,1,3,15,0,0,31
130 NEXT
```

Programa 6.2.

Para producir este sonido en el **AMSTRAD** dividimos el programa en dos partes. La primera es un bucle que produce un sonido con un valor de tono creciente. La segunda parte, correspondiente a la explosión, produce dicho efecto añadiendo ruido al sonido mediante el último parámetro de **SOUND** (el ruido tiene que estar en el rango 0-31).

El programa 6.3 simula un grito de muñeca en el **COMMODORE**:

```
10 REM *********************
20 REM * GRITO DE MUNECAS *
30 REM *     COMMODORE     *
40 REM *********************
50 PRINT CHR$(147)
60 PRINT SPC(212);"GRITO DE MUNECAS"
70 FOR I=54272 TO 54296
80 POKE I,0
90 NEXT I
100 POKE 54296,15
110 POKE 54276,65
120 POKE 54277,15
130 FOR I=200 TO 5 STEP -2
140 POKE 54273,40:POKE 54272,I
150 NEXT I
160 FOR I=150 TO 5 STEP -2
170 POKE 54273,40:POKE 54272,I
180 NEXT I
```

Programa 6.3.

El efecto se consigue con dos bucles decrecientes que producen la variación de las bajas frecuencias.

Otro ejemplo interesante es el sonido de una sirena.

```
 10 REM ************
 20 REM *   SIRENA   *
 30 REM * SPECTRUM *
 40 REM ************
 50 CLS
 60 PRINT AT 11,13;"SIRENA"
 70 FOR I=5 TO 15
 80 BEEP 0.01,I
 90 NEXT I
100 FOR I=15 TO 5 STEP -1
110 BEEP 0.01,I
120 NEXT I
130 GO TO 70
```

Programa 6.4.

Este sonido lo produce el **SPECTRUM** mediante dos bucles que se alternan, uno creciente y otro decreciente que producen la variación de la frecuencia del sonido.

El programa 6.5 produce el sonido de un helicóptero en vuelo.

```
10 REM ***************
20 REM * HELICOPTERO *
30 REM *     MSX     *
40 REM ***************
50 CLS
60 LOCATE 15,11:PRINT "HELICOPTERO"
70 FOR I=0 TO 13
80 READ D
90 SOUND I,D
100 NEXT
110 RESTORE
120 GOTO 70
130 DATA 200,15,200,15,200,15
140 DATA 0,7,23,23,23,255,1,12
```

Programa 6.5.

Este programa ha sido desarrollado en **MSX** por lo que es necesario manejar los 13 registros de **SOUND** para producir el sonido.

El sonido de una alarma también resulta interesante de cara al diseño de videojuegos.

```
10 REM ******************
20 REM * ALERTA GENERAL *
30 REM *     AMSTRAD    *
40 REM ******************
50 ENT 1,80,-4,1
60 CLS
70 LOCATE 16,13:PRINT "[ ALERTA !"
80 SOUND 1,478,50,15,0,1,0
90 GOTO 60
```

Programa 6.6.

El programa 6.6 reproduce el efecto de una alerta en AMSTRAD. Para que el sonido resulte más real se ha generado una envolvente de tono con una sección.

Finalmente el programa 6.7 produce el sonido de un grifo goteando en IBM.

```
10 REM ******************
20 REM * GRIFO GOTEANTE *
30 REM *      IBM       *
40 REM ******************
50 KEY OFF:WIDTH 40:CLS
60 LOCATE 13,13:PRINT "GRIFO GOTEANTE"
70 I=100
80 I=I+I^1.01
90 SOUND I,40/I
100 IF I<1000 THEN GOTO 80
110 SOUND 32767,30
120 GOTO 70
```

Programa 6.7.



MUSICA ALEATORIA

Finalmente vamos a desarrollar un programa para generar música aleatoria con el SPECTRUM.

El inconveniente de la música aleatoria es que no siempre resulta agradoble a los oídos, sin embargo siempre es una experiencia interesante.

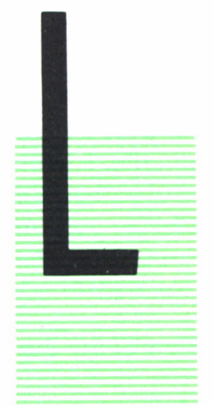
```
 10 REM ********************
 20 REM * MUSICA ALEATORIA *
 30 REM *     SPECTRUM     *
 40 REM ********************
 50 CLS
 60 PRINT AT 10,8;"MUSICA ALEATORIA"
 70 DIM N$(12,4)
 80 FOR I=1 TO 12
 90 READ N$(I)
100 NEXT I
110 DATA "DO","DO#","RE","RE#"
120 DATA "MI","FA","FA#","SOL"
130 DATA "SOL#","LA","LA#","SI"
140 RANDOMIZE 0: LET NOTA=0
150 LET N=INT (RND*8)
160 LET D=INT (RND*200)/100
170 LET S=INT (RND*2)
180 LET NO=NOTA
190 IF S=0 THEN  LET NOTA=NOTA-N: GO TO 210
200 LET NOTA=NOTA+N
210 IF NOTA<1 THEN  LET NOTA=NOTA+12: GO TO 210
220 IF NOTA>12 THEN  LET NOTA=NOTA-12: GO TO 220
230 PRINT AT 14,14;N$(NOTA);"   "
240 BEEP D,NO
250 GO TO 150
```

Programa 6.8.

El bucle FOR-NEXT de las líneas 80-100 almacena en una matriz los nombres de las notas de una escala cromática; dichos nombres los lee de las líneas DATA 110-130. En la línea 150 se genera un número al azar entre 0 y 7 que, según el valor que tome la variable S (0 ó 1), se sumará o restará a la variable NOTA determinando el tono que va a sonar. En la línea 160 se genera otro número al azar que determina la duración de la nota.

Hasta aquí hemos recorrido parte del camino en la experimentación del sonido con ordenador, queda mucho por descubrir, sigamos investigando.

EL COMANDO SOUND EN EL COMMODORE



LA capacidad de generación de sonido del COMMODORE sólo es posible a través de un montón de POKEs.

Los problemas son más graves si pensamos que el chip de sonido del COMMODORE exige un determinado orden en la ejecución de los POKEs. Por ejemplo, poner el volumen a cero inmediatamente después de una nota puede producir un «click» poco musical, mientras que seleccionar la forma de onda antes de activar el generador de envolvente no produce ningún sonido. Es mucho mejor, evidentemente, disponer de un único comando SOUND que se encargue de todo de una manera sencilla y eficaz.



SOUND

Sintaxis: SOUND (voz), (tono), (volumen), (A), (D), (S), (R), (forma de onda), (ciclo de trabajo).

Donde: (voz) selecciona una de las tres voces del SID, numeradas de 0 a 2.

(tono) define la frecuencia del oscilador seleccionado anteriormente entre 0 y 65535.

(volumen) define el volumen desde 0 (silencio) a 15.

(A) define el parámetro tiempo de ataque de la envolvente.

(D) define el parámetro tiempo de decaimiento de la envolvente.

(S) define el parámetro tiempo de sostenimiento de la envolvente.

(R) define el parámetro tiempo de relajación de la envolvente.

(forma de onda) selecciona el tipo de onda: 0=triangular; 1=diente de sierra; 2=rectangular, y 3=ruido.
(ciclo de trabajo) define el ancho del pulso o ciclo de trabajo de la onda rectangular. Sólo es aplicable si selecciona este tipo de onda. El rango efectivo es entre 0 y 4096.

SOUND no requiere siempre todos los parámetros. Por ejemplo, es posible obtener una nota sólo con SOUND 0,40000.

SOUND toma ciertos valores por defecto, ya que el chip SID requiere en general que se inicialicen todos los parámetros para cada sonido. Los valores por defecto para el comando mínimo SOUND (voz),(tono) son: forma de onda triangular, A,D,S,R igual a 0,9,0,0 y el máximo volumen. Observe que si se especifica uno cualquiera de los parámetros ADSR debe hacerse lo mismo con los otros tres, y que si se selecciona la forma de onda 2 (rectangular), debe especificarse la anchura de los pulsos. Después de producida la nota, se deja que el sonido se amortigüe libremente, lo que puede producir una vibración permanente para ciertas envolventes. En este caso el sonido deberá desactivarse después de cierto tiempo con un comando SOUND (voz) —por ejemplo, SOUND 0— para desactivar la voz 0. SOUND sin parámetros tiene el efecto de borrar todos los registros y «dormir» al SID después de un «click».

Trabajar con más de una voz no es problema. Simplemente ejecute dos o tres SOUND, uno después de otro. Por ejemplo:

```
10 SOUND 0,4461
20 SOUND 1,5619
30 SOUND 2,6675
```

tocarán un acorde en tono de DO.

Listado en ENSAMBLADOR:

```
1000 *=$8F00
1010 ;----------------
1020 WAVEF  =$0336
1030 VOLUM  =$0337
1040 VOIC   =$0338
1050 ;----------------
1060 ;--DESACTIVA TODOS LOS CANALES--
1070 ;SI NO HAY PARAMETROS BORRA TODOS
1080 ;LOS REGISTROS DEL SID
1090 ;----------------
1100 SOUN   BNE SOUN2
1110 ;----------------
1120        LDX #$18
1130        LDA #$00
1140 SOULP  STA $D400,X
1150        DEX
```

```
1160         BPL SOULP
1170         RTS
1180 ;---------------
1190 ;--OBTIENE EL NUMERO DE CANAL O VOZ
1200 ;RECHAZA VALORES MAYORES QUE 2/ BO-
1210 ;RRA EL REGISTRO DE FORMA DE ONDA
1220 ;DEL CANAL ELEGIDO
1230 ;---------------
1240 SOUN2  JSR $B79E
1250         CPX #$03
1260         BCS ERSOU
1270         STX VOIC
1280         LDA #$00
1290         LDY VOICB7,X
1300         STA $D404,Y
1310 ;---------------
1320 ;SI NO HAY MAS PARAMETROS SALE
1340 ;---------------
1350         JSR $0079
1360         BNE SOUN3
1370         RTS
1380 ;---------------
1390 ;SALIDA DE ERROR:"ILLEGAL QUANTITY"
1400 ;---------------
1410 ERSOU  LDX #$0E
1420         JMP ($0300)
1430 ;---------------
1440 ;--OBTIENE EL PARAMETRO FRECUENCIA-
1450 ;LO GUARDA EN LOS REGISTROS DEL CA-
1460 ;NAL ELEGIDO
1470 ;---------------
1480 SOUN3  JSR $AEFD
1490         JSR $AD8A
1500         JSR $B7F7
1510         LDX VOIC
1520         LDY VOICB7,X
1530         LDA $14
1540         STA $D400,Y
1550         LDA $15
1560         STA $D401,Y
1570 ;---------------
1580 ;--OBTIENE EL PARAMETRO VOLUMEN--
1590 ;SI NO SE ESPECIFICA, EL VALOR POR
1591 ;DEFECTO ES EL MAXIMO
1600 ;---------------
1610 DFOLT  LDA #$0F
1620         ORA $D418
1630         STA $D418
```

```
1640         JSR $0079
1650         BEQ DFOLT0
1660 ;----------------
1670         JSR $AEFD
1680         JSR $B79E
1690         STX VOLUM
1700         LDA $D418
1710         AND #$F0
1720         ORA VOLUM
1730         STA $D418
1740 ;----------------
1750 ;--OBTIENE LOS PARAMETROS DE LA
1760 ;ENVOLVENTE--
1770 ;SI NO SE ESPECIFICAN, LOS VALORES
1771 ;POR DEFECTO DE ADSR SON 0,9,0,0
1780 ;----------------
1790 DFOLT0 LDX VOIC
1800         LDY VOICB7,X
1810         LDA #$09
1820         STA $D405,Y
1830         LDA #$00
1840         STA $D406,Y
1850         JSR $0079
1860         BEQ DFOLT1
1870 ;----------------
1880         JSR ATKDCY
1890         LDX VOIC
1900         LDY VOICB7,X
1910         STA $D405,Y
1920         JSR ATKDCY
1930         LDX VOIC
1940         LDY VOICB7,X
1950         STA $D406,Y
1960 ;----------------
1970 ;--OBTIENE LA FORMA DE ONDA--
1980 ;EL VALOR POR DEFECTO ES 0 = TRIAN-
1990 ;GULAR / 1 = DIENTE DE SIERRA ,
1991 ;2 = PULSOS RECTANGULARES ,3 =RUIDO
2000 ;----------------
2010 DFOLT1 LDA #$11
2020         STA WAVEF
2030         JSR $0079
2040         BEQ DFOLT2
2050 ;----------------
2060         JSR $AEFD
2070         JSR $B79E
2080         TXA
2090         AND #$03
```

```
2100        TAX
2110        LDA WAVE,X
2120        STA WAVEF
2130        CMP #$41
2140        BNE DFOLT2
2150 ;----------------
2160 ;SI LA FORMA DE ONDA SON PULSOS,
2170 ;OBTIENE LA ANCHURA DE LOS MISMOS
2180 ;----------------
2190        JSR $AEFD
2200        JSR $AD8A
2210        JSR $B7F7
2220        LDX VOIC
2230        LDY VOICB7,X
2240        LDA $14
2250        STA $D402,Y
2260        LDA $15
2270        AND #$0F
2280        STA $D403,Y
2290 ;----------------
2300 ;--TOCA LA NOTA--
2310 ;ACTIVANDO EL BIT DE PUERTA DEL RE-
2320 ;GISTRO DE FORMA DE ONDA DEL CANAL
2321 ;ESPECIFICADO
2330 ;----------------
2340 DFOLT2 LDX VOIC
2350        LDY VOICB7,X
2360        LDA WAVEF
2370        STA $D404,Y
2380        RTS
2390 ;----------------
2400 ;--OBTIENE LOS VALORES ADSR--
2410 ;COMBINA DOS SEMIOCTETOS EN UN BYTE
2420 ;----------------
2440 ATKDCY JSR $AEFD
2450        JSR $B79E
2460        TXA
2470        ASL A
2480        ASL A
2490        ASL A
2500        ASL A
2510        STA WAVEF
2520        JSR $AEFD
2530        JSR $B79E
2540        TXA
2550        AND #$0F
2560        ORA WAVEF
```

```
2570         RTS
2580 ;----------------
2590 ;--TABLA CON LOS VALORES DE LAS
2591 ;FORMAS DE ONDA--
2600 ;----------------
2610 ;WAVE  .BYTE $11,$21,$41,$81
2620 ;----------------
2630 ;--TABLA DE CANALES (PARA INDEXADO)
2640 ;----------------
2650 VOICB7 .BYTE $00,$07,$0E
2660 ;----------------
```

Cargador BASIC

```
100 C=0
101 READ D$
102 IF D$="END" THEN PRINT D$:GOTO1000
103 D=VAL(D$):IF D<0 THEN 106
104 X=X+1:C=C+D:POKE 36607+X,D
105 GOTO 101
106 IF C=ABS(D) THEN 100
107 PRINT"ERROR EN EL BLOQUE";1+INT((X-.5)/64)
108 GOTO 100
110 REM --- BLOQUE 1
111 DATA208,11,162,24,169,0,157,0
112 DATA212,202,16,250,96,32,158,183
113 DATA224,3,176,17,142,56,3,169
114 DATA0,188,246,143,153,4,212,32
115 DATA121,0,208,6,96,162,14,108
116 DATA0,3,32,253,174,32,138,173
117 DATA32,247,183,174,56,3,188,246
118 DATA143,165,20,153,0,212,165,21
119 DATA-7176
120 REM --- BLOQUE 2
121 DATA153,1,212,169,15,13,24,212
122 DATA141,24,212,32,121,0,240,20
123 DATA32,253,174,32,158,183,142,55
124 DATA3,173,24,212,41,240,13,55
125 DATA3,141,24,212,174,56,3,188
126 DATA246,143,169,9,153,5,212,169
127 DATA0,153,6,212,32,121,0,240
128 DATA24,32,215,143,174,56,3,188
129 DATA-6885
130 REM --- BLOQUE 3
131 DATA246,143,153,5,212,32,215,143
132 DATA174,56,3,188,246,143,153,6
133 DATA212,169,17,141,54,3,32,121
134 DATA0,240,47,32,253,174,32,158
135 DATA183,138,41,3,170,189,242,143
136 DATA141,54,3,201,65,208,27,32
```

```
137 DATA253,174,32,138,173,32,247,183
138 DATA174,56,3,188,246,143,165,20
139 DATA-7870
140 REM --- BLOQUE 4
141 DATA153,2,212,165,21,41,15,153
142 DATA3,212,174,56,3,188,246,143
143 DATA173,54,3,153,4,212,96,32
144 DATA253,174,32,158,183,138,10,10
145 DATA10,10,141,54,3,32,253,174
146 DATA32,158,183,138,41,15,13,54
147 DATA3,96,17,33,65,129,0,7
148 DATA14
149 DATA-5147
150 DATAEND
1000 REM --ENCADENA EL NUEVO COMANDO--
1001 REM - SOUND
1002 POKE33337,255:POKE33338,142
1003 END
```

Programa A.1.

LA RUTINA SOUND

Selección de la voz o canal (líneas 1100-1370).

Si no hay parámetros, el bucle de las líneas 1120-1170 pone a cero todos los registros del SID, y después vuelve al BASIC. Si el número de canal especificado es mayor que 2, se rechaza y causa una salida de error a través de las líneas 1410-1420. Si el parámetro es legal, se guarda primeramente en VOIC y se utiliza luego para indexar la tabla VOICB7 (que contiene los posibles valores de canal multiplicados por 7) y borrar el registro de forma de onda de las líneas 1240-1300. Observe que se utiliza VOICB7 porque los registros del SID correspondientes a cada voz están separados 7 bytes. Así, por ejemplo, el registro de forma de onda de la voz 0 está en $d404+0; el de voz 1, en $D404 + 7, y el de la voz 2, en $D404 + 14. Si no hay más parámetros, se sale al BASIC.

Selección del tono (líneas 1480-1560)

Este parámetro se guarda en los registros de frecuencia de la voz correspondiente.

Selección del volumen (líneas 1610-1730)

Sólo hay un registro de volumen para las tres voces y, por tanto, simplemente guardamos el valor en el mismo. Este registro comparte otra fun-

ción, ya que los 4 bits de mayor peso seleccionan el filtro y, por consiguiente, debemos tener cuidado de no modificar estos bits. Este parámetro es opcional, y el valor por defecto es el máximo (15).

Selección de la envolvente (líneas 1790-1950)

Una única rutina, ATKDCY, en la línea 2440 guarda los valores de ataque/decaimiento y sostenimiento/relajación. Esta rutina obtiene 2 bytes entre 0 y 15 separados por comas, y los combina con el único byte. Como el volumen de estos parámetros son opcionales, siendo los valores por defecto 0,9,0,0 para A,D,S y R. Sin embargo, si se especifica uno de ellos, deben definirse todos los demás.

Selección de la forma de onda (líneas 2010-2140)

El valor por defecto es 0, onda triangular. Un 1 genera diente de sierra; un 2, una onda rectangular, y un 3, ruido. Si se elige la onda 2, rectangular, debe especificarse la anchura del pulso, o se producirá un error de sintaxis. Este valor puede ser hasta 65535, pero el rango efectivo es entre 0 y 4096, ya que el registro es de 12 bits.

Producción de una nota (líneas 2340-2380)

Una vez situados todos los parámetros, se produce la nota al escribir el valor de la forma de onda seleccionada en el registro correspondiente, y luego se sale al BASIC.

Una de las posibilidades más apasionantes de los ordenadores, aunque menos conocida, es la generación de sonidos.

Gracias a estas posibilidades, la composición musical deja de ser privilegio de unos pocos afortunados.

En este libro descubrirá cómo elaborar sonidos y aprovecharlos en la composición de melodías populares. Además, podrá convertir su ordenador en un instrumento musical, pudiendo dar rienda suelta a su imaginación sin haber estudiado solfeo.